ST. MARY'S UNIVERSITY FACULTY OF INFORMATICS

SHORT MESSAGE SERVICE SPAM DETECTION USING MACHINE LEARNING

By

FETIYA DINO

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

July, 2024

Addis Ababa, Ethiopia

ACCEPTANCE

SHORT MESSAGE SERVICE SPAM DETECTION USING MACHINE LEARNING

By

Fetiya Dino

Accepted by the Faculty of Informatics, St. Mary's University, In Partial Fulfillment of the Requirements for the degree of Master of Science in Computer Science

Thesis Examination Committee:

Internal Examiner

Álembante Mulu (Ph.D.)

DR. MESFIN ABEBE 22/07/24

External Examiner

Mesfin Abebe (Ph.D.)

Dean, Faculty of Informatics

Alembante Mulu (Ph.D.)

July, 2024

DECLARATION

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been duly acknowledged.

Fetiya Dino Hassen

Full Name of Student

_____

Signature

Addis Ababa

Ethiopia

This thesis has been submitted for examination with my approval as advisor.

Minale Ashagrie (Ph.D.)

Full Name of Advisor

_____

Signature

Addis Ababa, Ethiopia

July, 2024

Acknowledgment

I want to start by expressing my gratitude to the All-mighty Allah for his unwavering support and heavenly guidance during this study.

Next, I would like to sincerely thank Dr. Minale Ashagrie, my adviser, for his outstanding supervision, steadfast support, and ongoing encouragement during the study for my thesis. Their extensive experience, perceptive criticism, and kindness have been invaluable in helping me refine this thesis and advance as a researcher.

Finally, I want to express my gratitude to Husen Yusuf and Mohamed Dino in particular for their continuous support, love, and understanding. Throughout this journey, I have found strength and drive in their unwavering trust in me and support.

Fetiya Dino

Table of Contents

List of abbreviation

AdaBoost	Adaptive Boosting
CNN	Convolutional Neural Networks
DHMM	Discrete Hidden Markov Model
FPR	False Positive Rate
FP	False Positive
FN	False Negative
GRU	Gated Recurrent Unit
KNN	K Nearest Neighbors
LSTM	Long Short-Term Memory
MMS	Multimedia Messaging Service
MSISDN	Mobile Station International ISDN Number
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NSC	NUS SMS Corpus
OTP	One-time password
RNN	Recurrent Neural Networks
RF	Random Forest
SMS	Short Message Service
SVM	Support Vector Machine
SMproS	Small and Medium-Sized Enterprises Promotion Scheme
Sklearn	Scikit-Learn
TF-IDF	Term Frequency-Inverse Document Frequency
TP	True Positive
TN	True Negative
TP-PID	Transactional Processor Payment Identification

List of Figures

List of Table

Abstract

The usage of mobile phones has deeply integrated in society's modern life. Short Message Service (SMS), as a prevalent and cost-effective mode of telecommunication, is currently among the most extensively used methods of communicating with one another. But this ease of use has also led to the growth of SMS spam, which seriously jeopardizes the dependability and integrity of mobile communication. To solve this issue, we suggested a machine learning-based solution for effectively distinguishing genuine "ham" communications from malignant "spam" ones in the SMS communication space. The techniques use the SMS Spam Collection dataset and machine learning classifiers such as M-NB, SVM, KNN, RF, and AB algorithms to categorize short messages as ham or spam. The machine learning-based spam detection approach demonstrated impressive performance, demonstrating how well it works to detect messages that are spam in communications on mobile devices. The careful data preprocessing and feature engineering steps were instrumental in building a robust and accurate spam detection model. Thoroughly cleaning and transforming the SMS collection data through techniques like removing stopWords, punctuation, text normalization and feature selections were crucial for preparing the SMS dataset to be effectively leveraged by the machine learning algorithm. These data preparation and feature engineering efforts were essential for overcoming the unique challenges of SMS data to create an effective spam detection algorithm that can recognize unsolicited SMS messages on mobile devices. After implementing and evaluating such proposed models, our evaluation performance measures yielded remarkable results, with the SVM model emerging as the top performer in the ML-based spam detection system with 98.3% accuracy, 100% precision, 96% recall, and 91% F1-score.

Keywords:

SMS, Spam Detection, Machine Learning, Natural Language Processing, NB, SVM, KNN, RF, AdaBoost.

Chapter 1 : Introduction

1.1 Background of the study

There is an exponential tendency in the rise in the use of mobile devices [1-3]. The usage of short messages (SMS) on mobile devices has significantly increased as a result of the swift development of technology and the growing popularity of content-driven advertising. According to the Ericsson mobility report November 2023 there is more than one connected device per person, since the estimated human population is around 8.5 billion people [4]. Users' reliance on mobile devices has risen as a result of their broad use in daily life and mobility. These days, users keep private and sensitive data on mobile devices, including email addresses, contact lists, banking details, and other private data [5] [6]. Furthermore, the fast advancement of mobile phones enhances day-to-day tasks by enabling instantaneous contact and exchange of information. Text messaging services like short messages (SMS) are compatible with landline phones as well as smartphones. SMS traffic has dramatically grown as a result.

Several people and businesses have utilized SMS throughout the years for sending instant messages for things like one-time passwords (OTPs) and transactions in money. 2.2 trillion messages in the form of SMS and MMS were sent by US cellphone owners in 2020 [7]. Because more people are using text messages on their phones, spammers are more interested in, and the volume of spam has exploded because text messages are a very effective form of communication that does not require the internet. From the huge amount of text messages, we get every day; spam refers to unsolicited, often automated messages that are irrelevant or unwanted by the recipient. These can include phishing attempts, advertisements, or other malicious content. In contrast, "ham" messages are legitimate, desired communications from trusted sources.

Furthermore, today's generation of mobile phones use beyond their original purpose as communication tools For example, keeping private data for the purpose of creating and maintaining financial transactions, shopping lists, notes, and papers, among many other

uses. As a result, a significant number of users are inundated with spam SMS messages. In order to distribute spam to gain financial or commercial benefit, spammers are using this form of communication to gather sensitive information, such as credit card numbers. Spammers are individuals or businesses that send unsolicited messages. They bombard people with messages for personal or professional purposes. For instance, Scammers are collecting users' credit card information and using it to send spam communications aimed at financial or commercial gain. Spam messages can trick cell phone users into revealing their personal information, with serious consequences [9] [10]. Malicious people are interested in hacking the mobile phone because a huge amount of information would be available on it, some of which is personal, is stored on the device. When the phone is hacked, the hacker can access the device and all its data without the user's knowledge. As a result of that personal information can be compromised [10]. Now that the issue is so bad, a proper spam prevention solution is required.

Although there are many SMS spam filtering systems available [11]- [18]. It is necessary to use the latest methods to solve this problem. Since the message must be sent according to the communication standard, text classification techniques are needed to identify communications as spam or ham. In general, spam filtering work is a binary classification issue where every SMS messages must be identified as spam or raw.

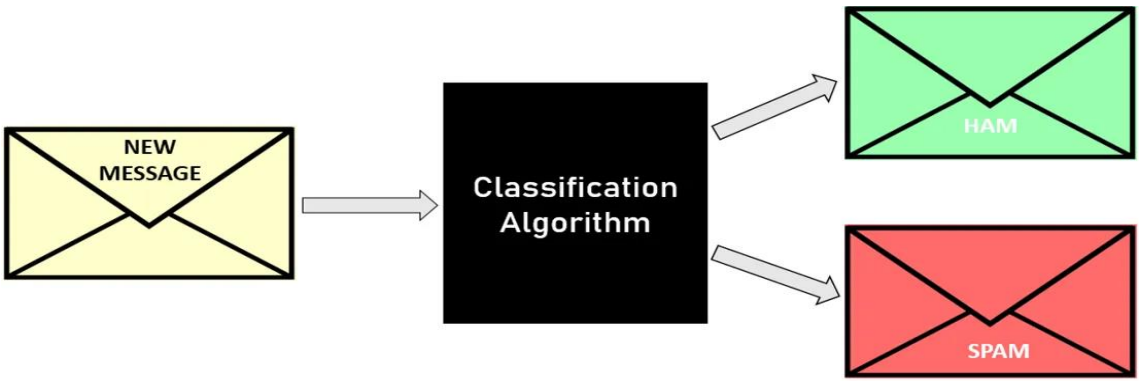

Figure 1.1: Machine learning system for detecting SMS spam.

With the use of large SMS collecting datasets containing both valid and spam messages, these machine learning algorithms for classification and detection have been taught to recognize the minute patterns and traits that set spam apart from authentic texts.

From analyzing word frequencies and message structures to scrutinizing sender information and metadata, machine learning models are capable of crafting intricate decision boundaries that can reliably flag potentially malicious messages. Now days, machine learning will continue to improve. As vast dataset becomes available, machine learning algorithms will become even more advanced and effective at SMS spam detecting. Hence, by deploying powerful classification and detection techniques, we can give users better control over their mobile messaging. This will help create a future where people can communicate securely and without interruption from spam.

Experiment with various machine learning classification models to identify the best effective strategy with high accuracy. To fully evaluate the effectiveness of SMS filtering out spam techniques, it is required to study a range of critical performance assessment metrics and statistics. In order to improve the security and dependability of mobile communication for users, the spam detection system can be optimized and ensured to reliably distinguish between legitimate and malicious SMS messages with the help of the primary metrics such as true positive rate and false positive rate, which offer further insight through the model's strengths and weaknesses. In order to do this, the model should be able to properly detect spam without mistakenly reporting an excessive number of genuine messages, as evidenced by an increased rate of true positives and a small number of false positives. Researchers may obtain a more comprehensive knowledge about each model's capabilities by examining a range of performance indicators, including accuracy and recall, which are additional metrics that might be helpful.

1.2 Statement of the Problem

Users of mobile phones frequently have issues with spam texts. Due to its affordability and ease of use, SMS has grown in popularity as mobile communication has progressed and as mobile phones have been more widely used. As we use SMS for many purposes, we receive many messages and notifications in our inboxes. Messages can be difficult to control and manage, and the frequency of spam messages makes the problem worse, causing victims

to lose money or other important information. The most worrisome message is spam, and many people find it annoying to receive these unwanted messages. Spam attacks can cause serious problems with our personal, financial, and other sensitive information [5] [10]. Clicking on a fake link can result in loss of important information or even spam deleting the real message, ignoring the real information. We may encounter some spam related issues for the following reasons: i.e., 'Families in Need'; 'You won'; and 'You Have Money Back' are some of the most well-known spam messages [19][20]. In theory, it is nice to receive an unexpected gift.

On the other hand, getting a notification that you won an SMS contest you didn't participate in is purely a phishing attempt. This is a common tactic used by scammers to try to gain access to your personal information or get you to pay a fee. Phishing scams exploiting SMS messaging have become increasingly prevalent, as they can leverage the ubiquity of mobile devices and the inherent trust people often place in text communications. If you are not sure the claim is legitimate, it is crucial to call the company directly to double-check and do not respond or provide any information, as this could lead to the compromise of your sensitive data or financial loss.

The world of mobile banking has revolutionized the way we manage our financial affairs, providing convenience and accessibility like never before [21] [22]. At the core of this revolution are two primary modes of communication between the customer and the bank: push mode and pull mode. SMS banking is an application for providing banking and financial services via text messaging. Customers can use their mobile devices to send messages to the bank. One of two types of work; push mode or pull mode is used with SMproS. When using the push model, the mobile customer sends a service request SMS to the bank and vice versa. Your mobile banking password has been changed for security reasons and all your SMS payments are spam. Fraudsters can dishonestly deceive financial institutions such as customers to obtain important personal and financial information [9].

Moreover, we get tons of text messages every day. Whether it's spam or legitimate, it can be difficult to approve all incoming SMS. Manually reviewing and categorizing every

incoming SMS is simply not feasible, as it is an arduous and time-consuming process prone to human error and inconsistency. This overwhelming volume of daily text messages underscores the critical need for robust and automated SMS spam detection solutions. Therefore, the primary goal of the present study is to separate spam from legal SMS using machine learning techniques. Algorithms for ML classifiers including AB, SVM, KNN, RF, and M-NB. The capacity of the algorithms to automatically identify patterns and characteristics from big datasets is one of the benefits of using machine learning-based SMS spam detection paradigms. This allows for flexible and scalable approaches that can keep up with the always changing spam strategies. Machine learning techniques can adjust their categorization skills when spammers come up with new ways to evade detection.

1.3 Research Questions

The following research questions are the focus of this work:

- In what ways might mobile consumers benefit from the categorization and implementation of natural language processing (NLP) approaches for SMS spam detection?
- Which machine learning method is more effective in identifying spam SMS messages?
- How can the effectiveness of the suggested SMS spam detection methodology be assessed?

1.4 Objectives of the Study

1.4.1 General Objective

The main objective of this research project is to use machine learning techniques to develop a model for SMS spam detection.

1.4.2 Specific Objectives

In order to accomplish the study's overall objective, the following specific goals need to be met:

- To review literature related to SMS spam filtering techniques and methods.
- To identify various SMS spam filtering methods
- To identify basic components of SMS spam detection model
- To collect and preprocess datasets for the purpose of SMS spam detection.
- To train the model using the experimental data sets and machine learning algorithms
- To evaluate the proposed model.

1.5 Significance of the study

This research developed an effective SMS spam detection machine learning algorithm that can detect and will let people know about spam; thereby helping individuals or businesses reduce the number of spams. The core of this research work involves using machine learning algorithms to analyze the content and patterns of incoming SMS messages. In order to train the ML algorithms to identify the traits of spam, significantly larger datasets including both known spam and genuine communications would be used for training. To do so, this research work is important for individuals or companies that want to be protected from spam text. Therefore, we propose a solution for the distribution of unwanted SMS spam groups.

1.6 Limitation and Scope of the study

The aim of this effort is to develop machine learning based SMS classification algorithms that can accurately discriminate between spam and ham texts. This research work's primary goal is to use a ML model that can assess a message's legitimacy or spam status based on its context and content. The scope of this study includes in-depth research of information and intelligence distribution to identify the most appropriate machine learning methods for developing distribution patterns for SMS collection messages. The research must also include the collection and analysis of such a large SMS data to inform and validate the

model. Research limited to text messages and not cover other forms of communication channels such as emails or social media. Additionally, the scope of this study does not include analysis of the legal or ethical aspects of SMS distribution, but only focuses on the process of establishing the distributed standard. Overall, the capability of the research method aims to ensure that the research remains focused and controlled, while at the same time allowing in-depth research into important issues and problems.

1.8 Organization of the study

Our remaining work is arranged as follows: The literature review and associated activities are covered in chapter II. Chapter III delves into the study methodology, including an investigation of the study's design, target population, instances, and sampling methodologies. Consideration was also given to suggested algorithms, data sources, techniques, instruments, and data processing techniques. Chapter IV presents the study's findings and commentary. Chapter V concludes with the study's results, suggestions, and future research directions.

Chapter 2 : Literature Review and Related Work

2.1 Introduction

In recent decades, research on spam detection and categorization has been increasingly popular. Like email spam, the issue of SMS spam can be tended to through legislative, business, or innovative means. This interest has significantly increased since the emergence of ML and its algorithms. Researchers have explored various machine learning techniques, such as NB, DT, SVM, K-NN, RF, and AB algorithms, to build robust SMS spam detection models. NLP techniques also have enhanced SMS spam detection by enabling deeper analysis of message content. The NLP-based analysis is important because it helps the spam filters stay effective even as spammers change their tactics over time. Research and innovation are still being done to provide comprehensive and flexible spam detection systems since spammers are always changing their strategies. This section covers pertinent research on SMS spam identification and categorization methods that has been done by different investigators in the corresponding categories.

2.2 Rule-based Approaches

The Rule-based frameworks are the earliest, simple, and easy to implement, and most straightforward techniques for detecting SMS spam rely on rule-based manners. These methods entail formulating a predefined set of guidelines and heuristics that may be applied to recognize possible spam communications. The authors [23] suggest that the proposed framework can be integrated into existing mobile security solutions to enhance protection against phishing attacks. In this study work, the authors evaluated real-world SMS messages, including both legitimate and smashing messages and they achieved an accuracy of over 95% in detecting phishing messages, which is outperforming traditional machine learning-based techniques. By incorporating the changing nature of spam features, the authors [24] presented a novel method for effective and reliable SMS spam detection utilizing a separate hidden Markov model (DHMM). This work also can be integrated into mobile security solutions or SMS filtering services to enhance protection against SMS-based phishing and spam attacks. The suggested method, the DHMM, performed better in

terms of F1-score, recall, and accuracy, so it was a viable option for practical use. However, the limitation of rule-based spam detection is that it can't keep up with how spammers evolve their tactics. As spammers get more sophisticated, they find ways to bypass the predefined rules, like using tricky language or constantly changing their sender information. High false positive rates—in which valid communications are mistakenly classified as spam—may result from this. To address these shortcomings, researchers have explored more advanced machine learning-based techniques [25] [26] [27] [28].

2.2 Machine Learning Algorithms

Unlike rule-based solutions, machine learning-based SMS spam detection uses data-driven algorithms to automatically identify patterns and characteristics that differentiate spam from valid texts. These methods usually include training a classification model, such as a NB, a SVM, or DT. The authors [25] showed the efficiency of machine learning techniques in solving the spam detection challenge. They demonstrate that both models SVM and RF outperform in both email and SMS spam detection tests. In [11], the authors suggested a bi-level solution that combines text classification and clustering techniques to effectively filter SMS spam and identify threads. This bi-level method can improve SMS spam detection accuracy by utilizing supervised machine learning techniques. In this work, the authors used an SVM model and attained an accuracy of 97.2%, outperforming the Naive Bayes classifier. In [26], the authors believe that machine learning-based approaches beat traditional rule-based and keyword-based spam detection methods in terms of overall performance when addressing the SMS spam detection problem. They performed exceptionally well with SVM, Random Forest, and Logistic Regression, with 98.4%, 97.2%, and 97.8% accuracy, respectively. The authors [29] investigated the application of SVM to categorize SMS messages as spam or ham in the telecoms business. When compared to other techniques such as NB and DT, the SVM model outperformed them in the SMS spam detection job, achieving an accuracy of 97.1% on the SMS dataset, as well as high precision (97.6%) and recall (96.7%) in recognizing spam messages.

Moreover, the advantage of machine learning approaches is their ability to adaptively improve their performance as they are exposed to more data. As new spam tactics emerge, the models can learn to recognize the evolving patterns without the need for manual rule-

based updates, making them more robust and accurate over time. However, the ML-based SMS spam detection approaches are the need for a large dataset and the computational complexity of training and deploying the models, which can be more demanding than rule-based approaches. To overcome this gap researchers proposed hybrid and deep learning-based SMS spam detection techniques [14] [30] [27] [15] [28] [31] [32].

2.3 Hybrid Approaches

Researchers have looked on hybrid systems that mix the two methods in order to strengthen the advantages that exist between rule-based and ML-based approaches. The rule-based filter quickly identifies obvious spam messages using a set of predefined rules. The remaining messages are then analyzed by the more advanced machine learning model, which can handle complex and evolving spam patterns. In [14], the authors proposed a hybrid SMS spam filtering framework that combines multiple machine learning algorithms. In this study, the authors demonstrate that, when measured in terms of accuracy, precision, recall, and F1-score, the suggested hybrid SMS spam filtering system performed better than the separate machine learning models. It results in discomfort, security problems, and monetary losses. The study [30] also proposes hybrid approaches that use ML techniques for both spam identification and sentiment analysis of SMS texts. In both instances, the writers showed excellent recall, accuracy, precision, and F1-score.

2.4 Deep Learning Approaches

The discipline of SMS spam identification has come a long way in the last few years thanks to the development of deep learning algorithms. These methods retrieve complex linguistic and semantic elements from the message content by using sophisticated neural network designs, such as recurrent neural networks (RNN) and convolutional neural networks (CNN). These deep learning models excel at detecting SMS spam by learning complex message patterns, allowing them to adapt to evolving spam tactics better than traditional rule-based approach and classical machine learning techniques. The authors of [15] used CNN and RNN models to investigate how deep learning approaches affected the job of SMS spam filtering. They then compared the models' performance to that of traditional machine learning methods like SVM and NB classifiers. To do so, they achieved accuracy

up to 97.2%, precision up to 96.8%, and recall up to 97.5% - improvements of 5-7 percentage points over the traditional methods. The performance of several deep learning models and machine learning methods for the task of SMS spam identification is compared by the authors in [28]. Here, the authors showed that Random Forest, out of all the machine learning models, had the best accuracy of 92.4% in identifying SMS messages as either spam or ham (regular). Additionally, the LSTM network beat the CNN and GRU models in the deep learning techniques, with an accuracy of 94.7%. In comparison to the machine learning techniques, the deep learning LSTM model demonstrated better accuracy, recall, and F1-score overall.

A CNN and an LSTM network are combined in a unique optimization approach called the hybrid deep learning model, which is presented by the authors in [31]. They got excellent precision (97.5%) and recall (98.1%) scores, as well as substantial performance gains over previous deep learning systems for SMS spam filtering, with classification accuracy reaching 98.2%. The authors [32], Developed a deep learning-based system for automatically classifying SMS texts as spam or legitimate. They experimented with CNN, RNN, and a hybrid CNN-RNN model. In this investigation, the hybrid CNN-RNN model had the highest accuracy (96.8%) in categorizing SMS messages as spam or ham. In (28), the authors conducted a survey of several text classification algorithms, including their application to the challenge of detecting SMS spam. Additionally, they talked about how CNN and RNN fared better at classifying SMS spam than traditional machine learning algorithms. Overall, promising results have been shown in correctly detecting and categorizing SMS spam using both traditional ML and more sophisticated DL-based algorithms. Technique selection seems to be influenced by things like processing capacity, dataset properties, and the particular needs of the SMS spam detection system.

Table 2.1 presents a helpful summary of the most recent methods for detecting SMS spam, showcasing the variety of approaches investigated and the performance indicators that go along with them. This data might be a useful resource for practitioners and academics who are trying to improve SMS spam detection skills. The development of text message spam detection methods serves as a baseline for this study.

Table 2.1 The revision of the related works and their approaches

Title	Methodology	Accuracy	Author
Rule-Based System for Smishing Message Identification in a Mobile Setting	Rule-based approach	95%	[23]
A discrete hidden Markov model for SMS spam detection	Discrete Hidden Markov Model (DHMM)	Superior performance metrics	[24]
Machine Learning for SPAM Detection.	SVM and RF	98%, 97.2%	[25]
Employing bi-level text classification as well as clustering algorithms, SMS filtering of spam and thread identification are achieved.	SVM model, text classification, clustering techniques	97.2%	[11]
Detecting SMS Spam Through Machine Learning	SVM, RF, and LR	98.4%, 97.2%, 97.8%	[2]
Machine Learning and Deep Learning Methods for SMS Spam Detection	CNN and RNN	97.2%	[15]
An automated SMS spam categorization technique using deep learning: Applying learning algorithms to native datasets,	CNN, LSTM	98.2%	[31]
Text Classification Algorithms: A Survey	CNN, RNN	96.8%	[32]

An innovative optimization approach for message spam detection based on deep learning models	SVM, RF, CNN, LSTM, GRU	92.4% for RF and 94.7% on LSTM	[28]

Chapter 3 : System Model and Research Methodology

The suggested machine learning algorithm for SMS spams detection and categorization is presented in this section. We describe from data collection to model section and raining thought the research methodology and workflow followed in developing the solution, including the data pre-processing schemes and dataset visualization techniques employed. We next go into more depth about the suggested machine learning-based solution for classifying and detecting SMS spam, including the precise methods and techniques employed. In conclusion, we cover the fundamental ideas of machine learning and its applications, the algorithms and classification methods that form their basis and the suggested machine learning-based method for classifying and detecting spam in SMS messages, along with its implementation and hyper parameter tuning. The choice of SMS spam detection technique is often influenced by factors like dataset characteristics, computational resources, and system requirements, as certain approaches may be more suitable for specific application scenarios.

3.1 Proposed System Architecture

The proposed approach for SMS detection of spam and categorization issues is demonstrated in Figure 3.8. Initially, we load the dataset after downloading the SMS spamming collected from the ML public repository. Next, we undertake a comprehensive data preprocessing stage. This involves techniques such as text cleaning, remove stopWords, text normalization, and tokenization to transform the raw SMS message data into a format suitable for feature extraction and model training. This preparatory step is crucial in ensuring the data is optimized for the subsequent machine learning algorithms. We first preprocess the SMS messages and then extract pertinent information from them.

The feature engineering process for SMS spam detection models often involves a variety of techniques to extract relevant input features from the raw SMS collection text datasets. The bag-of-words model, which depicts text as an arranged group of words, as well as TF-IDF (Term Frequency-Inverse Document Frequency), a statistical measure reflecting a word's significance to a document in a corpus, are two popular methods. Beyond these basic text representation methods, more advanced natural language processing techniques

may also be employed to capture deeper semantic and syntactic information from the SMS content. Ongoing research also continues to explore novel feature engineering and modeling techniques to further improve the state-of-the-art in this domain, with the goal of developing increasingly robust and accurate SMS spam classifiers. Simple or complicated, the chosen characteristics function as variables of input for the machine machine learning algorithms that are employed in SMS detection for spam. Since careful feature engineering and selection are essential elements in creating trustworthy and precise SMS spam classifiers, these feature extraction and selection processes in this research study have a major influence on the effectiveness of the detection of spam system. To help the machine learning algorithms create useful prediction models, the chosen features must efficiently capture the traits that set spam apart from real communications.

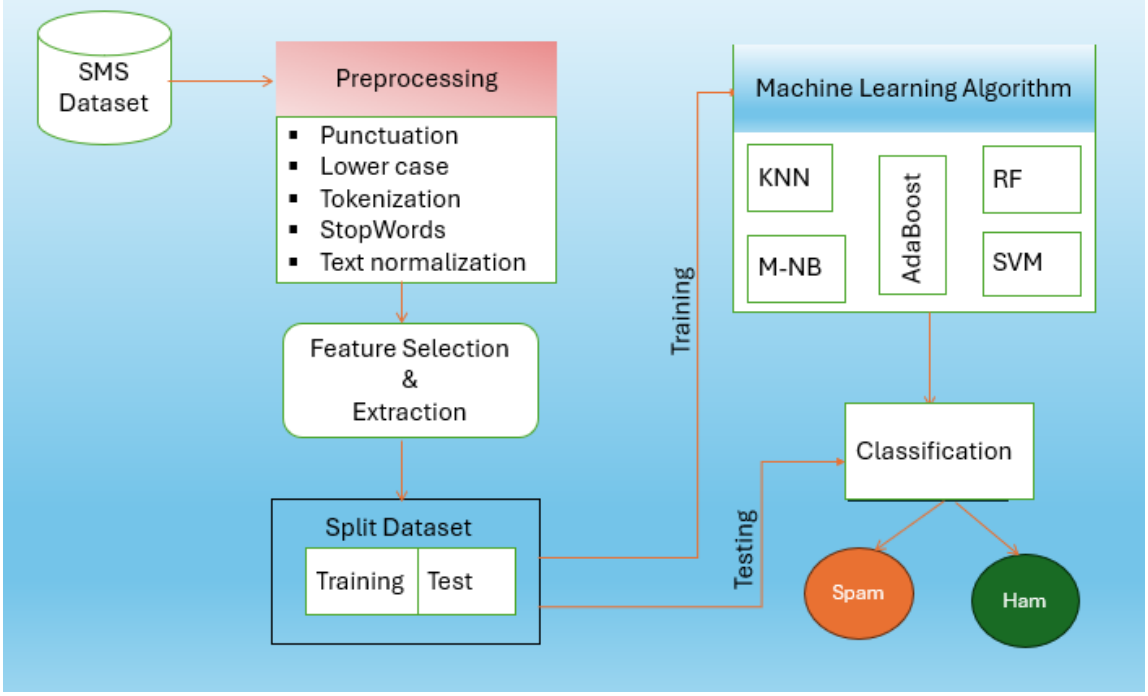

Figure 3.1 Proposed system architecture

We divided this data set into subsets for training and testing following the preprocessing phase. This separation makes it possible to assess the model's performance objectively because the testing set isn't viewed during the training process. We are finally undertaking of a model selection and training process and carefully evaluate, and compare the performance of the proposed learning algorithms, such as M-NB, SVM, K-NN, RF, and

AB. Utilizing feature engineering approaches such as the bag-of-words model and TF-IDF, these algorithms are able to extract pertinent features from the SMS text. We use assessment measures like accuracy, precision, recall, and F1-score to assess each model's performance once it has been trained on the prepared training data.

3.2 Dataset Collection

A popular dataset for the classification of text and spam prevention algorithms that need to identify messages sent via SMS as spam or ham is the SMS spam collection. The set of 5574 text messages that we used to collect SMS spam was sourced through the machine learning open repository. It includes 5,574 English text messages, 4,827 ham SMS messages and 747 spam messages, among a wide assortment of legitimate and spam communications.

Table 3.1 Number of SMS spam collection

Ham	**Spam**	**Total**
4,827	747	5574

Only 13.4% of the messages in the spam SMS collection dataset are spam; the remaining 86.6% are ham or legislative communications. The graph that follows pie chart displays the dataset's distribution.

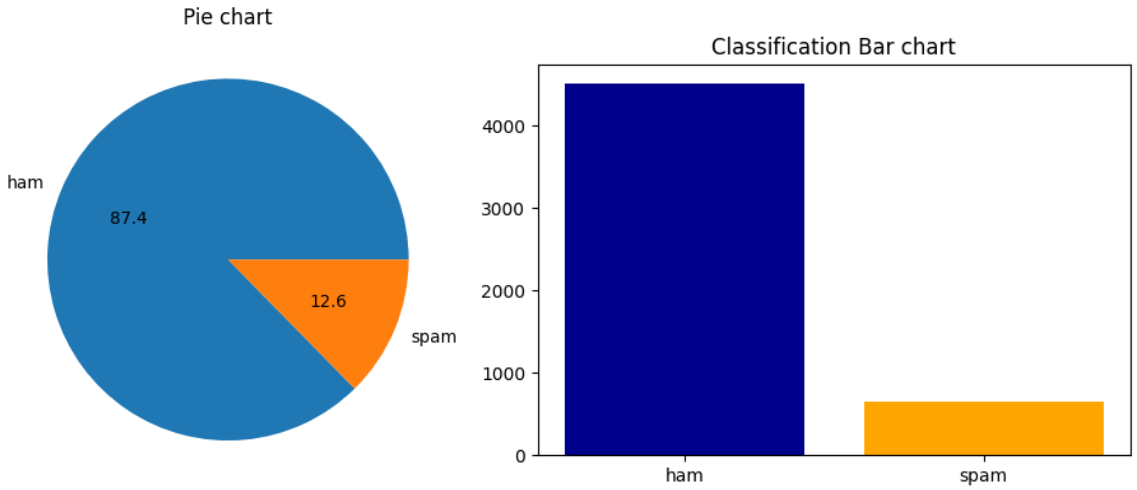

Figure 3.2. Comparison SMS Spam collection dataset both class count and percentage

The datasets are organized as CSV entries, separated by commas. There is one text for every line in these archives. There are two fields on each line: v1 contains the message label (spam or ham), and v2 contains the raw text contents.

index	target	text
0	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives around here though
5	spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, å£1.50 to rcv
6	ham	Even my brother is not like to speak with me. They treat me like aids patent.
7	ham	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune
8	spam	WINNER!! As a valued network customer you have been selected to receivea å£900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.
9	spam	Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030

Figure 3.3 Sample data set used

3.3 Data Pre-Processing

In order to provide cleaned and processed data that can be used to train and assess various machine learning classification models, preprocessing—the act of eliminating redundant and unnecessary data—is essential to the machine learning pipeline. In this instance, the preprocessing task's objective is to produce a reliable and comprehensive dataset that can be utilized for training models that use machine learning to reliably discern between spam and valid SMS messages—even when noisy or unorganized text information is present.

The following summarizes the main data pre-processing stages associated with the text messages Spam Acquisition dataset:

1) The Dataset consists of two fields: a string representing the raw text message and a label designating the kind of communication (spam or ham). We carry out normalization and clearing:
    a) Eliminating non-textual features such as email addresses, URLs, and special characters.
    b) Changing every word to lowercase.
    c) Managing slang, acronyms, and spelling variants. Contractions are becoming more expansive (e.g., "don't" to "do not," "can't" to "cannot").
2) Most English stop words that don't have much significance, such as "the," "a," and "is," should be eliminated. These words may be downloaded from NLTK. This aids in lowering the feature space's dimensionality, one of the most common preprocessing techniques used in text retrieval tasks.
3) We simplify the words by stemming and lemmatizing them to their base or root form. Lemmatization takes a more complex method depending on the morphology of the word, whereas stemming follows rules.
4) Tokenizing and vectorizing text input using the scikit-learn (Sklearn) library's CountVectorizer (Bag-of-Words) function. The textual data is transformed via the vectorization process into a numerical representation that may be fed into machine learning models. The generated vectors, which are frequently used to

mathematically represent text data, incorporate the occurrence information of the words. The crucial actions are as follows:

a) Build a CountVectorizer class instance.
b) To construct the vocabulary (a dictionary of distinct terms) from the given text corpus, use the fit() technique.
c) To encode each text sample (such as a line of message) as a numerical vector, call the transform () method.
d) The resultant vector is as long as the terms, and each member indicates how many times the matching word appears in the supplied text.

5) The most significant or pertinent characteristics from the vectorized text will then be chosen using choice of features or dimensionality reduction.
6) We divided the dataset into two groups: a pair of trains (X_train, y_train) & a test (X_test, y_test) that were chosen at random and had a percentage of 70:30 (from the entire dataset).
7) We input the machine learning models, such as MNB, SVMs, KNN, RF, and AB, the data sets X_train, X_test, y_train, and y_test.

3.3.1 Stop Words, Lowercasing and Punctuations

Stop words are terms that are designed to be ignored by search engines, both during the indexing process for searching entries and during the retrieval of those items in response to a search query. With the NLTK library, you can use the Python code that follows to see all English stop words:

```
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
print(stopwords.words('english'))
```

['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what',

'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]

Then, we apply the techniques which is removing **StopWords** to improve the performance and increase classification accuracy, since meaningful tokens left from the given text messages. **Lowercasing** is converting all text messages to lowercase English alphabet. This ensures consistency and helps in treating words with different cases as the same. The python implementation code as follow:

```
df[‘text’] = df[‘text’].apply(lambda x: x.lower())
```

The other concept is "Remove punctuation marks (!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~) from the text. Punctuation may not always contribute to the meaning of words and can be safely

```
import string
df[‘text’]=df[‘review_text’].apply(lambda  x:x.translate
(str.maketrans(‘’, ‘’, string.punctuation)))
```

Take the words "I HAVE A DATE ON SUNDAY WITH WILL!" as an example. and text following the elimination of stop words, lowercase letters, and punctuation: "have date Sunday will"

3.3.2 Text Normalizing

The two steps of stemming and lemmatization combine to create a word's single canonical form. Using a list of frequently occurring prefixes and suffixes, the text normalization process known as "stemming" eliminates a word's beginning or end. It is a simple rule-based procedure that eliminates suffixes from every word, such as "ly," "ing," "es," "s," and so on. Lemmatization, on the other hand, is a methodical, systematic process that yields a word's root form. It is based on morphological analysis—the study of word structure and grammatical relationships—and vocabulary, or the meanings of words as found in dictionaries.

```
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
ps.stem('loving')  // return “love”
```

3.3.3 Tokenization and frequency calculation

It entails dividing the input material into more manageable, semantic chunks, including words, phrases, sentences, or even individual letters. By doing this, the text is ready for additional processing and analysis. Generally speaking, spam communications are lengthier and contain more letters and words than ham ones.

```
import nltk

# num of characters, ## num of words, and ### num of sentence respectively

df['num_characters'] = df['text'].apply(len)

df['num_words'] = df['text'].apply(lambda x:len(nltk.word_tokenize(x)))

df['num_sentences']= df['text'].apply(lambda x:len(nltk.sent_tokenize(x)))
```

	target	text	num_characters	num_words	num_sentences
0	0	Go until jurong point, crazy.. Available only ...	111	24	2
1	0	Ok lar... Joking wif u oni...	29	8	2
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2
3	0	U dun say so early hor... U c already then say...	49	13	1
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1

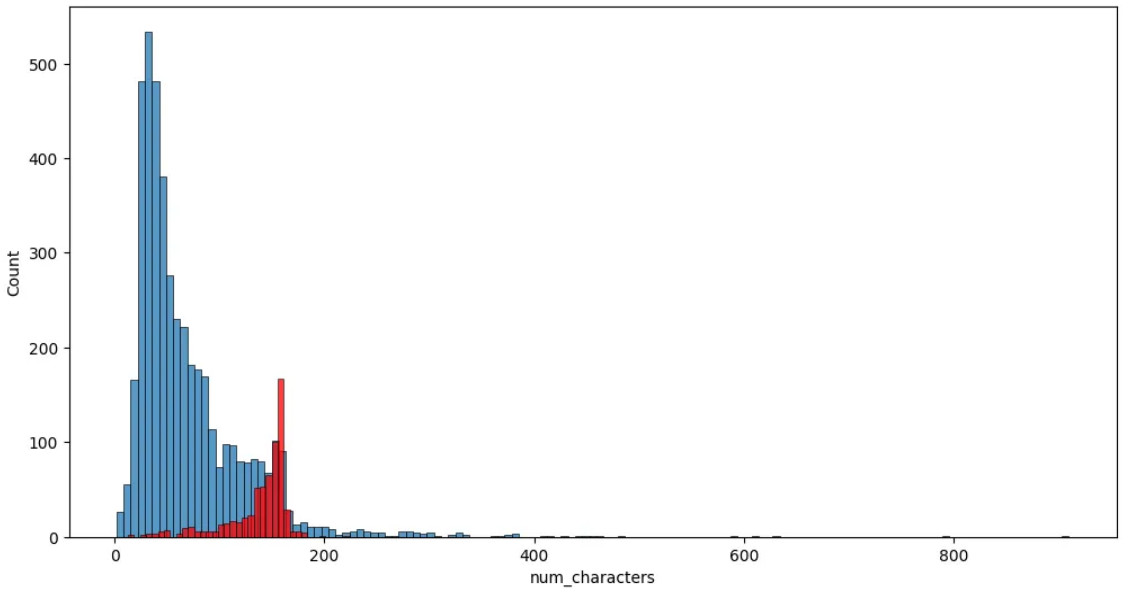

Figure 3.4. Message length distribution with respect to class.

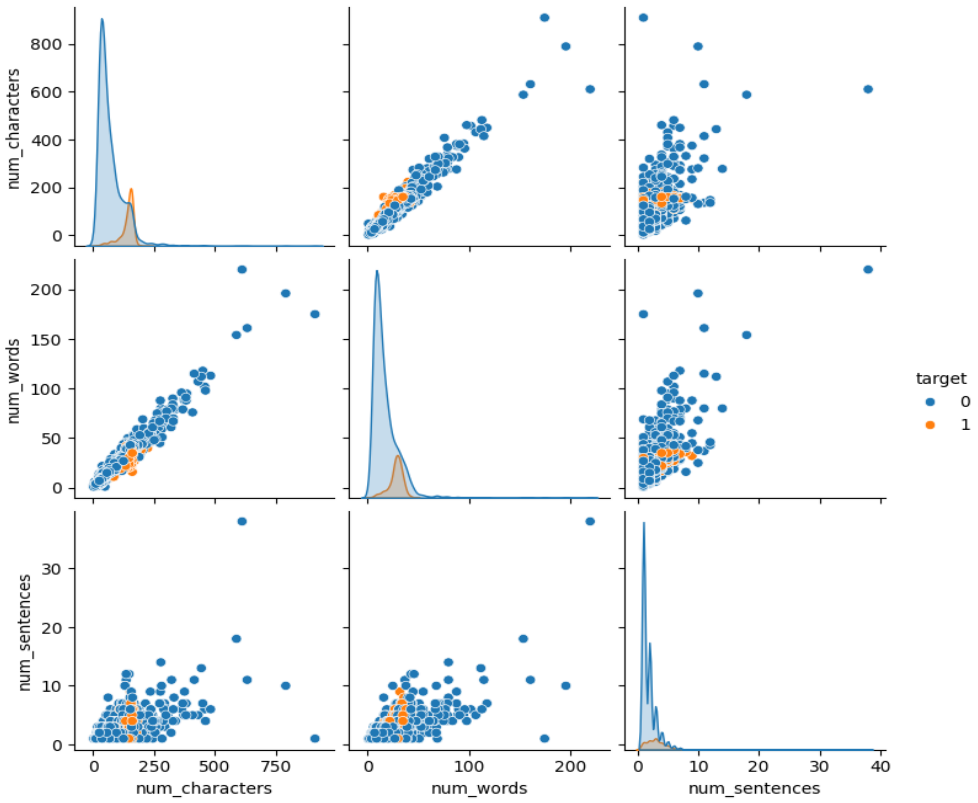

Figure 3.5 Message length distribution with respect to both classes.

3.4 Word Clouds for message visualization

Word clouds can be a visually engaging way to analyze and visualize the text, especially for problems like SMS spam detection.

```
from wordcloud import WordCloud
wc=WordCloud(width=500,height=500,min_font_size=10,background_color='white')
df['transformed_text'] = df['text'].apply(transform_text)
```

```
Spam_wc
= wc.generate(df[df['target']==1]['transformed_text'].str.cat(sep=" "))
plt.figure(figsize=(15,6))
plt.imshow(ham_wc)
ham_wc
= wc.generate(df[df['target']==0]['transformed_text'].str.cat(sep=" "))
plt.figure(figsize=(15,6))
```

```
plt.imshow(ham_wc)
```

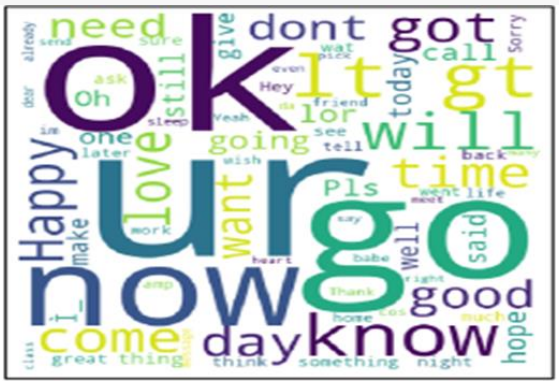

Figure 3.6(a): Spam Word Cloud

Figure 3.6(b): Ham's Word Cloud

3.5 Text vectorization

Text vectorization is the next step after using these preprocessing methods. It is the process of turning numerical vectors from textual data. All that the machine learning approach can handle is numerical data.

```
from collections import Counter
Counter(spam_corpus).most_common(15)
```

Result:

```
[('call', 320), ('free', 191), ('2', 155),
 ('txt', 141), ('text', 122), ('u', 119),
 ('ur', 119), ('mobil', 114), ('stop', 104),
 ('repli', 103), ('claim', 98), ('4', 97),
 ('prize', 82), ('get', 74), ('new', 64)]
```

```
from collections import Counter
Counter(ham_corpus).most_common(15)
```

Result:

```
[('u', 883), ('go', 404), ('get', 349),
 ('gt', 288), ('lt', 287), ('2', 284),
```

('come', 275), ('got', 236), ('know', 236),
('like', 234), ('call', 233), ('time', 219),
('ok', 217), ('love', 216), ('good', 213)]

```
from collections import Counter
spam_corpus = []
for msg in df[df['target'] == 1]['transformed_text'].tolist():
    for word in msg.split():
        spam_corpus.append(word)
bp = pd.DataFrame(Counter(spam_corpus).most_common(30))
plt.bar(bp[0],bp[1])
plt.xticks(rotation="vertical")
plt.show()
```

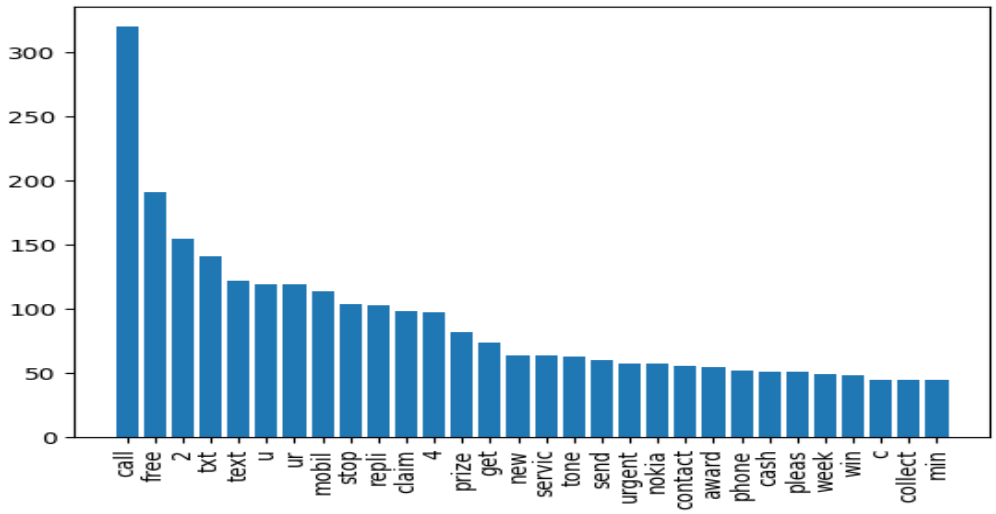

Figure 3.6 Most used words in spam messages.

```
from collections import Counter
ham_corpus = []
for msg in df[df['target'] == 0]['transformed_text'].tolist():
    for word in msg.split():
spam_corpus.append(word)
bp = pd.DataFrame(Counter(ham_corpus).most_common(30))
```

```
plt.bar(bp[0],bp[1])
plt.xticks(rotation="vertical")
plt.show()
```

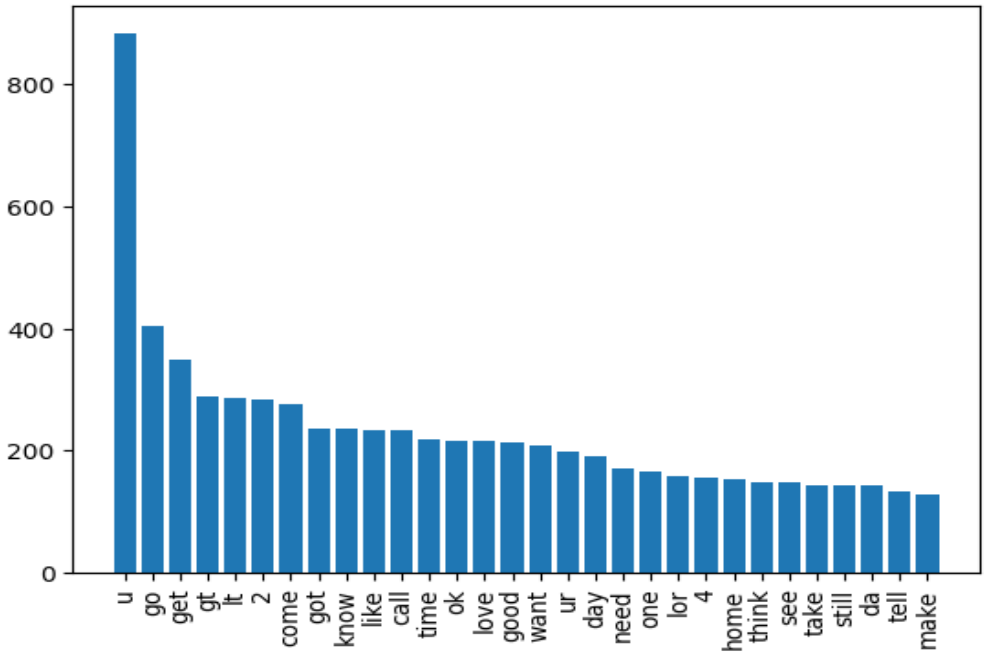

Figure 3.7 Most used words in ham messages.

The result on figure 3.7 and 3.8 shows the most frequently occurring words in the spam and ham messages corpora, which can provide insights into the different language patterns used in spam and legitimate messages. The insights gained from these word frequency visualizations can be a valuable input to the model selection and training process, as the researchers can leverage this information to inform feature engineering and selection of most appropriate ML algorithms for the task of SMS spam detection.

3.6 Feature extraction and Model

In order to train the machine learning model, features will need to be extracted from the preprocessed data. Techniques like bag of words and inverse document frequency, or TF-IDF, were used in this. TF-IDF functions nicely. To create a feature vector, a Bag of Words method just counts the instances of every phrase in the messages. In contrast, TF-IDF gives each word a weight determined by how frequently it appears in the message in relation to how frequently it appears in the entire corpus. All things considered, the goal of this

research is to address the ongoing problem of SMS spam by creating a precise machine learning model that can identify text messages as either ham or spam. The template scripts shown below are used to apply these feature extraction approaches.

```
From sklearn.feature_extraction.textimportCountVectorizer,TfidfVectorizer
cv = CountVectorizer()
tfidf = TfidfVectorizer()
X = tfidf.fit_transform(df['transformed_text']).toarray()
Y = df['target'].values
```

We concentrated on dividing the samples into sets for training and testing after the early stages of feature extraction and data preparation. This is a critical stage in the ML process since it enables an objective assessment of the model's effectiveness on hypothetical data. We use 20% of the data for testing and 80% of the dataset for training. In the world of ML phenomena, this 80/20 split is a frequently employed strategy because it strikes a decent compromise between the volume of the training set and the dependability of the test set. In the end, this data splitting process helps to design an efficient SMS spam detection technique by providing a solid and objective assessment of the models.

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=2)
```

We implement and assess the suggested ML models which could correctly categorize the messages as either spam or ham after dividing the available data into training and testing sets. The M-NB, SVM, KNN, RF, and AB algorithms were investigated in this study.

```
from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn import svm
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
From sklearn.metricsimport accuracy_score,confusion_matrix,precision_score
classifiers = {
```

```
        SVC': svm,
        'KNNeighbors' : knn,
        'Multinomial Naive Bayes' : mnb,
        'Random Forest' : rf,
        AdaBoostClassifier: ada
}
classifiers.fit(X_train,y_train),  predict (X_test)
```

We used the relevant classes and methods provided by the Scikit-learn library, a popular Python ML framework, to develop these algorithms. The training data are fitted for each machine learning model using classifiers.fit (X_train,y_train)). We then assess (predict (X_test)) each model's performance on the testing data by computing metrics like accuracy, precision, recall, and F1-score. Through extensive testing, we were able to determine the advantages and disadvantages of each method, which helped us, choose the best strategy for the SMS spam detection assignment. Later in chapter four, the specific experimental outcome and comments will be provided.

3.7 Machine Learning Classification Techniques

3.7.1 Machine Learning Basics

Without explicit programming, computers may learn and develop on their own thanks to machine learning. It is tuning things (data) into numbers and finding patterns in these numbers. The data can be anything (images, texts, videos, audio files, etc...) converted into numerical representations, and then machine learning identifies the patterns and insights within those numerical patterns. Machine learning can be advantageous for problems with complex rules, continuously changing the environments. Its adaptive nature allows machine learning models to learn new scenarios, overcoming the limitations of rigid, rule-based /traditional approaches. Additionally, machine learning excels at discovering insights within large collections of data. By identifying patterns, trends, and relationships that may not be easily discernible to the human eye, machine learning algorithms can uncover valuable insights within a hug dataset. This makes machine learning a powerful

tool for data analysis and knowledge extraction, especially in domains where the volume and complexity of data exceeds the capabilities of traditional analytical methods. The authors [33], recognize the critical role that machine learning plays in enabling systems to learn and adapt from data, moving beyond rigid, programmed algorithms.

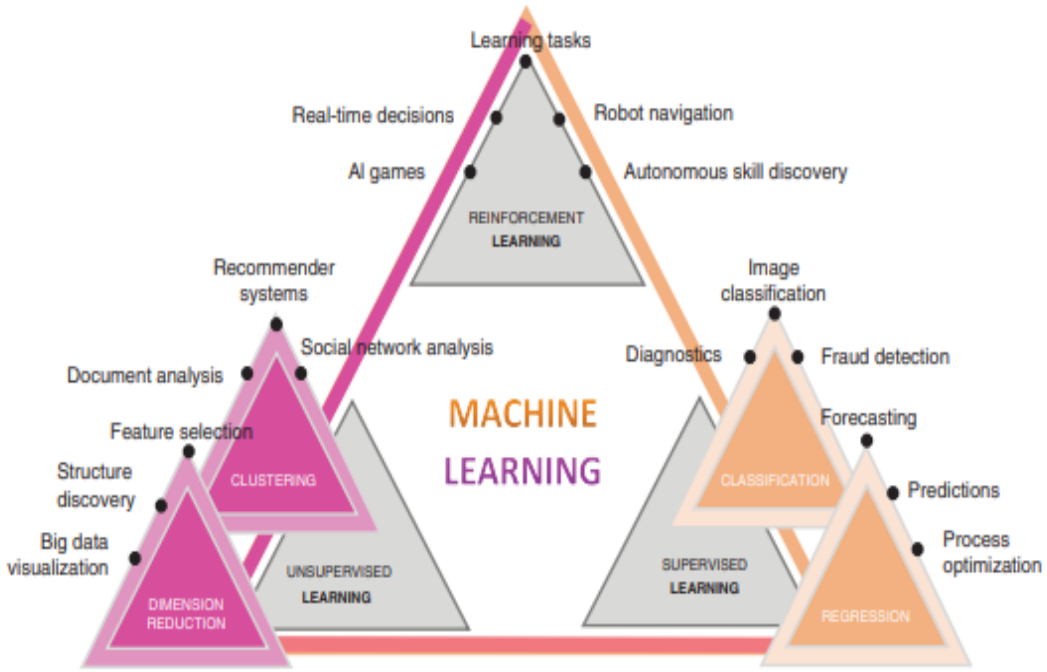

Figure 3.8 Taxonomy of Machine learning [33]

AI includes machine learning, which is the subset of AI that consists of methods that let computers understand data and provide AI applications. Despite the natural intelligence displayed by people and animals, which is demonstrated by machines, artificial intelligence, also known as machine intelligence, may be comprehended by intelligence. It examines methods of creating intelligent systems and gadgets that may ingeniously solve issues that are frequently viewed as human prerogatives. AI, then, denotes a machine that mimics human behavior in some way.

Deep learning, also known as deep neural learning or deep neural network is a branch of machine learning that draws inspiration from the architecture and functioning of the human brain. It employs multilayered artificial neural networks to assess a wide range of variables in a manner like to that of a human neural system. Its networks are capable of autonomously

learning from unlabeled or unstructured input. CNN and RNN, two deep learning models, are trained on a wide range of data, including text, audio, and pictures. They have demonstrated strong performance in a range of classification tests and the ability to understand intricate patterns across large datasets. In summary, machine learning is a branch of AI that helps computers learn from data, deep learning is a potent method for machine learning utilizing deep neural networks which has become a dominating approach in many applications, and artificial intelligence (AI) is the wide discipline concerned with creating intelligent systems [34- 41].

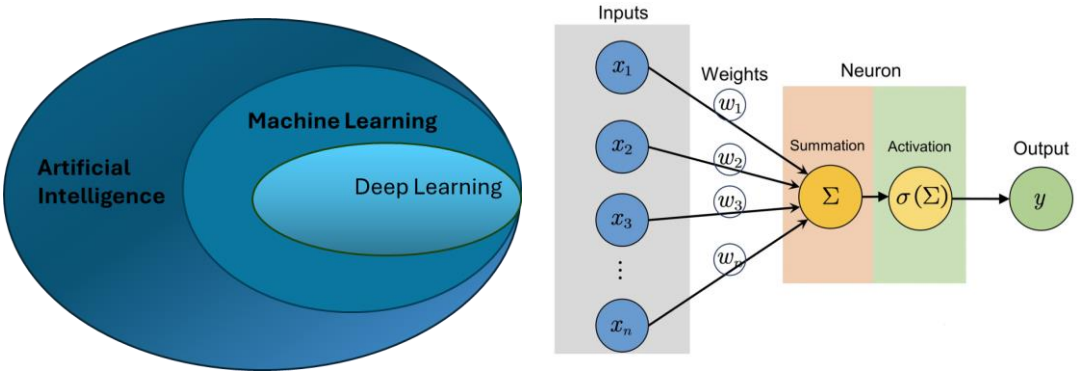

Figure 3.9 AI, Machine learning and Deep learning paradigm (left), Neural network model (right)

As shown in figure 3-2 (right) the simple neural network model diagram; the neural network as a black box, then we input data, after the neural network black box processing, and then output our data. It is like the human body is composed of several neurons, and from the mathematical theory, more neurons and more complex neural network architecture, can be more complex data processing.

3.7.2 Machine Learning Algorithm for Classification

Supervised learning together with unsupervised learning are the two primary paradigms of machine learning. Considering a labeled set of input-output pairs, the objective of the supervised training paradigm is to develop an algorithm that maps input characteristics to output targets. Regression tasks and classification tasks can have discrete or continuous output objectives. The goal of a regression method is to develop an algorithm which will forecast an ongoing numeric output variables supplied any number of input characteristics. The most prevalent regression algorithms include linear regression, which represents the

output as a linear combination of the input characteristics, and more complicated nonlinear regression approaches like polynomial regression DT, and NN-based regression models. The purpose of algorithms for classification is to train a model that can predict an input instance's class label, whereby the classes labeled are distinct groups [35] [36]. As a result, classification algorithms based on machine learning are employed to forecast the class or categories within which a newest finding belongs, using a training set of observations labeled by class. Its goal is to learn how to map input characteristics to final class labels, which may subsequently be used to generate predictions about fresh, unknown data. Some of the widely utilized ML classification methods are:

A. **Logistic Regression**

Logistic Regression is employed to address probabilistic statistical model classification challenges in machine learning [42]. Although this technique is applicable to regression and classification issues alike, it is most commonly used for binary classification tasks. In such tasks, the objective is to predict whether an instance falls into one of two categories (ham/spam SMS, tumor/non-tumor, buy/not buy, dog/cat, etc.). Any input value may be mapped to a number that ranges from 0 to 1, which can be understood as the likelihood that the instance belongs to either spam or ham (in our example). This function is also known as the sigmoid function or logistic function. For example, categorizing a person's likelihood of purchasing a product according on their income and age. Age (years), income (dollars/ETB), and purchase (binary; 0 if the item was not purchased, 1 if it was).

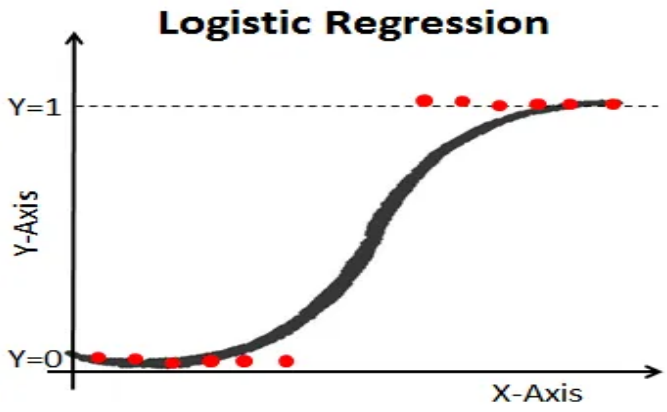

Figure 3.10. Logistic regression for classification

This ML classification model is able to correctly identify incoming SMS messages as "spam" or "ham" (non-spam) in our instance.

B. Naive Bayes

Operating under the premise that characteristics are independent of each other, the method known as Naive Bayes is based on the Bayes theorem [43] [18]. This approach may be used in a variety of real-world contexts, including spam filtering and document or text categorization, including binary and multi-class categories. As a result, it is a very good choice for SMS spam identification and classification, where it is possible to calculate the likelihood that a class (spam or ham) would exist given the input data as follows:

$$P(class|features) = (P(features|class) * P(class)/P(features) \ldots\ldots\ldots \quad (3.1)$$

Assume that the objective of Y is to categorize an SMS message as "spam" (S) as well as "ham" (H) based on attributes denoted as X = {x1, x2,...). The Naive Bayes method chooses a class with the highest probability after calculating the subsequent likelihood of the message either spam or ham using Bayes' theorem. The following is the mathematical formulation:

The likelihood that a message is spam (P(S)) and the likelihood that it is ham (P(H)). Calculate the conditional probabilities of the features given the classes:

P(x1|S), P(x2|S), ..., P(xn|S): Probabilities of the features given the message is spam.
P(x1|H), P(x2|H), ..., P(xn|1): Probabilities of the features given the message is ham.

Calculate the posterior probabilities using Bayes' theorem:

P(S|X) = (P(x1|S) * P(x2|S) * ... * P(xn|S) * P(S)) / P(X)
P(H|X) = (P(x1|H) * P(x2|1) * ... * P(xn|H) * P(H)) / P(X)

Classify the message based on the higher posterior probability:

If P(S|X) > P(H|X), classify the message as **spam.**
If P(S|X) > P(H|X), classify the message as **ham.**

For such a kind of spam filtering classification problem (SMS spam), multinomial Naive Bayes is Suitable since the features represent word frequencies or counts in documents and each feature (word) is treated as a discrete event with a count. Nevertheless, in contrast to other classification methods like LR, SVM, or deep learning models, it might not be more suitable for more complicated spam detection jobs.

C. K-Nearest Neighbor

KNN is an easy-to-understand algorithm that uses the labels of a new instance's nearest neighbors in its feature space to classify it. It is a grouping directed learning calculation[18]. It predicts the name of the class as new information and uses something like its contributions to the preparation set.

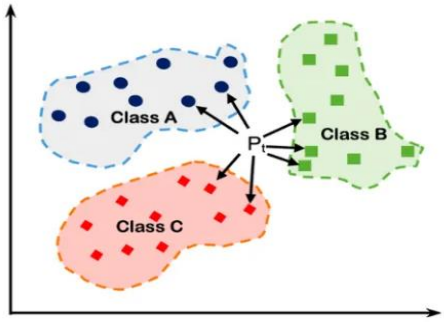

Figure 3.11 K-Nearest Neighbor for classification [44].

In our case, this machine learning classification model is like the Naive Bayes approach, the first step is to preprocess the SMS messages and extract relevant features and classify a new, unseen SMS message, through the follow's steps:

- Represent the new message as a feature vector inside the same multidimensional feature space.
- Calculate the distance (e.g., Euclidean distance) between the new message's feature vector and the feature vectors of all the training instances.
- Determine which K training examples have the shortest distances to the feature vector of the new message.

- In accordance with a majority vote from the K nearest neighbors, classify the new communication as spam or ham.
  - A message is labeled as spam if the bulk of its K closest neighbors consist of spam.
  - The new message is categorized as ham if the bulk of its K closest neighbors are ham messages.

While the KNN algorithm can be a useful tool for SMS spam detection, these shortcomings highlight the need for a more comprehensive and adaptive approach to tackle the challenges of real-world spam detection. Combining the KNN algorithm with other machine learning techniques or incorporating domain-specific knowledge can help resolve these issues and raise the spam detection system's overall efficacy.

D. Support Vector Machine

A reliable machine learning technique that is frequently used for both regression and classification problems is called Support Vector Machine (SVM). It operates as an algorithm for supervised learning, defining a decision boundary by finding the hyperplane that best divides data points into distinct classes. By maximizing the distance among the selected margin and the support vectors—the nearest data items from each class—the SVM method aims to achieve its goal. SVM is well known for its proficiency with high-dimensional data and its ability to generalize well to new data. It is extensively utilized in many different domains, including spam filtering, picture and classification of text, handwritten digit identification, cancer diagnosis, and bioinformatics studies [29].

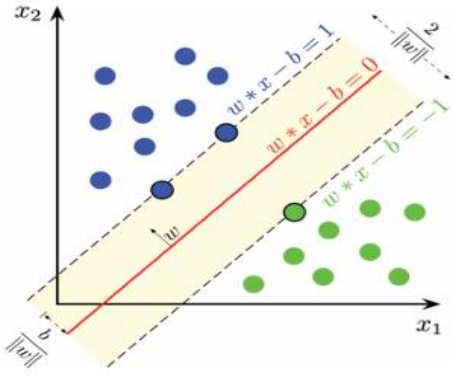

Figure 3.13 for binary classification using support vector machines [45].

The task at hand involves identifying the best hyperplane to distinguish between spam and ham messages from a dataset of SMS messages. Specifically, this involves maximizing the margins between the hyper plane along with the closest points of data (support vectors) and decreasing the misclassification errors, which can be expressed by the slack variables. Each message in the dataset is expressed as a feature vector X and labeled Y either as "spam" (y = 1) or "ham" (y = 0). P instances (xi) with labels (yi) provide the input for the training algorithm.

$$(x_1, y_1), (x_2, y_2), (x_3, y_3), \quad . \quad . \quad . \quad . \quad (x_p, y_p) \tag{3.2}$$

$$\text{where} \quad \begin{cases} y_k, = 1, & if\ x_k\ \epsilon\ class\ A \\ y_k, = -1, & if\ x_k \epsilon\ Spam \end{cases}$$

During a learning phase, the variables that comprise of the decision functions D(x) are established for this training. Next, using the following rule, the categorization of unidentified patterns is predicted:

$$X \epsilon A\ if\ D(x) > 0$$

$$X \epsilon B\ otherwise$$

The dot product of two vectors with a linear classification algorithm is defined as $u^T x = \sum_i u_i x_i$. For a linear discriminant function classifier.

$$f(x) = u^T x + b \tag{3.3}$$

Where the following gives the decision threshold function:

$$D(x) = \sum_{i=1} u_i\ \varphi i(\mathrm{x}) + b$$

where b is the bias, φi are predetermined functions of x, and vectors u is the weight of the vector. If b = 0, every point in a hyper plane are orthogonal to u, thus the hyper plane goes through the origin as a result of the bias shifting the hyperplane farther from the origin.

They create two halves of the room. The classifier's decision boundary is the line that divides the regions into positive and negative classifications. The decision boundary function's representation in the dual space is provided by:

$$D(x) = \sum_{k=1}^{p} a_k K(x_k, x) + b \quad (3.4)$$

E. Random Forest

A popular ensemble learning technique that works well for regression as well as classification applications is called Random Forest. Decision tree structures and the idea of ensemble learning serve as the mathematical cornerstones of this approach. In our scenario, a majority vote among all the decision trees' forecasts determines whether the input should be classified as "spam" or "ham".

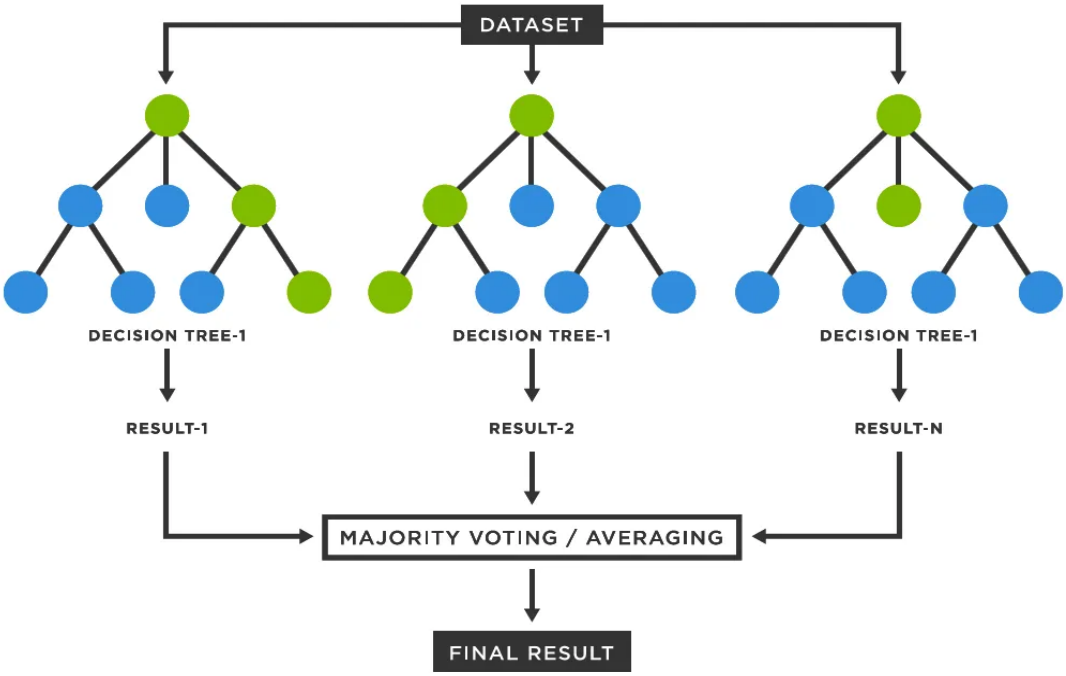

Figure 3.12 Random Forest for classification [46].

F. AdaBoost

AdaBoost is a powerful ensemble machine learning algorithm that has significantly contributed to SMS spam detection and classification and beyond. It is a boosting algorithm

designed to improve the performance of weak learners by combining their predictions into a robust and accurate model. AB achieves this by iteratively adjusting the weights of the data points, focusing more on those that were previously misclassified. This adaptive weighting process allows the algorithm to focus on the most challenging aspects of the classification task, enhancing its overall predictive accuracy. As a result, AB is not only effective in filtering spam but is also widely utilized in various domains that require sensitive and precise classification capabilities.

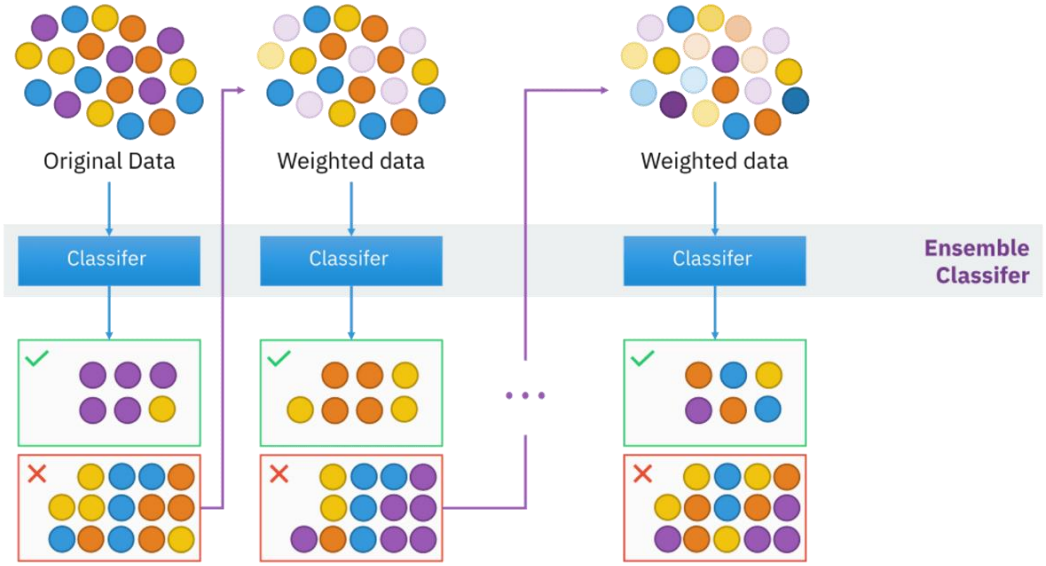

Figure 3.13 Adaptive Boosting for classification [47].

In this instance, a different well-liked machine learning classification technique may be used to solve the SMS spam detection and categorization problem with the contributions that follow:

- ➔ Ability to handle multi-dimensional feature spaces.
- ➔ Robustness to noise and outliers in the data
- ➔ Automatic feature selection and importance ranking
- ➔ Adaptability to employ diverse weak learners (such as decision stumps and decision trees)
- ➔ Interpretability of the final classifier as a weighted combination of weak learners

3.8 Evaluation Measures

We used common assessment measures for classification tasks, such as accuracy, precision, recall, and F1-score, to evaluate the performance of the suggested model. We also utilized the confusion matrix, which is a commonly used tool that displays these four metrics. These metrics are essential for assessing a classification model's efficacy and precision because they provide information on the model's capacity to discriminate among spam or ham messages, a critical distinction for applications such as SMS spam filtering.

Table 3.2. Evaluation metrics

Measurement	**Description**
TP	Instances in which the model accurately predicted the class to be 1 (Spam), even when the message's actual class is 1 (signaling that it is spam). This indicates that the model correctly recognized the spam mails.
TN	Instances in which the model accurately predicted the class to be 0 (Ham), even when the message's actual class is 0 (showing if the message is Ham). This indicates that the model recognized the Ham messages correctly.
FP	Instances in which the model mistakenly predicted the class to be 1 (Spam) while the message's true class was 0 (Ham). This indicates that a ham message was mistakenly classified as spam by the model.
FN	These are the instances when the model forecasted the category as 0 (Ham) but in fact the message's real class was 1 (Spam). This indicates that the model was unable to recognize a message as spam.

Accuracy: The ratio of correctly anticipated observations to total observations is known as accuracy. When assessing classification models, it's crucial to take the dataset's

distribution into account, despite the common belief that the algorithm with the greatest degree of precision is the best. Since the data set is preferably symmetric or slightly skewed, it is a trustworthy measure. But accuracy can be deceptive when distributions are highly skewed. As such, it's critical to take into account extra measures in order to fully assess a model's performance.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.5}$$

False alarms, or FP Rate (FPR), are defined as the proportion of negative instances that were mistakenly forecasted as positive. It is computed as the ratio of real negative observations to incorrectly projected positive observations.

$$FPR = \frac{FP}{FP + TN} \tag{3.6}$$

Precision: The ratio of genuine positive observations to all expected positive observations is known as precision. It provides an answer to the following query: of all communications marked as spam, what proportion are truly spam? A low FPR is correlated with good accuracy.

$$\text{Precision } = \frac{TP}{TP + FP} \tag{3.7}$$

The fraction of positive instances that are mistakenly labeled as negative, or spam messages that are mistakenly classed as ham, is known as the FN Rate (FNR).

$$\text{FNR } = \frac{FN}{FN + TP} \tag{3.8}$$

Recall (Sensitivity): The ratio of accurately anticipated positive findings to all realized positive observations—including both TP and FN—is known as recall (sensitivity). It provides an answer to the following query: of all SMS which are genuinely spam, what proportion have been classified as such? Reduced false negative rate is linked to high recall.

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3.9}$$

The FPR calculates the percentage of FP in the given samples, whereas precision calculates the chance where the positive classification is accurate. When FP mistakenly classifies a spam message as ham, it may result in the communication being automatically filtered and ultimately destroyed without the user's awareness. On the other hand, FN, which labels a spam communication as ham, wastes time because the user needs to read, remove, and perhaps expose the spam message.

Chapter 4 : Experiment Result and Discussion

4.1 Introduction

This section includes the datasets, setup settings, experiment findings, and performance evaluations of all suggested algorithms. The suggested machine learning algorithm model was assessed using its identification rate (recall), precision, F-measure, and accuracy—well-known standard assessment metrics for the classification algorithms. In conclusion, we present the suggested algorithms' total comparative performance based on these assessment measures for key performance indicators.

4.2 Software tools and configuration setting

A set of potent software tools inside the Google Colab platform, as indicated in table 4.1, were carefully chosen and integrated to enable the results of our experiments on SMS spam detection by machine learning approaches. Each of these tools played a crucial role in streamlining our experimental workflow, enhancing the effectiveness of your data processing and model development efforts. Scikit-learn is the core ML library in our technical stack. It is a widely used and well-documented framework for Python. The modular design and extensive utilities provided by Scikit-learn allowed us to seamlessly integrate it into your Colab-based workflow, enabling efficient model development, tuning, and evaluation. In this research work, we noted the critical role played by the NLTK, a comprehensive complement of Python libraries for working with human language data. NLTK proved invaluable in the text preprocessing stage of our SMS spam detection experiment, equipping us with a flexible set of tools for tasks such as **tokenization**, **stopWords** removal, and **normalizing**.

By integrating NLTK into the Colab environment, we were able to streamline the data preparation process, ensuring that the SMS text data optimized for subsequent modeling efforts. Colab's cloud-based Jupiter Notebook environment provided a collaborative and easily accessible workspace, enabling us to manage the entire process, in an efficient manner, from data preparation to model building and assessment. The ability to quickly

iterate on our experiments and leverage pre-installed libraries and dependencies within Colab streamlined our workflow, allowing me to focus on the core objectives of our thesis work.

Table 4.1 Dataset, software tool and configuration parameter

<table>
<tr><th colspan="3">Datasets</th><th colspan="3">Configurations / hyperparameters</th></tr>
<tr><td>Message type</td><td>No of SMS</td><td>% of SMS</td><td>Model</td><td>Parameter</td><td>Setting</td></tr>
<tr><td>Ham</td><td>4827</td><td>86.6 %</td><td>SVM</td><td>kernel</td><td>linear</td></tr>
<tr><td>Spam</td><td>747</td><td>13.4 %</td><td>SVM, RF</td><td>random_state</td><td>42</td></tr>
<tr><td rowspan="2">Total</td><td rowspan="2">5574</td><td rowspan="2">100%</td><td>AdaBoost, RF</td><td>n_estimators</td><td>100</td></tr>
<tr><td>KNN</td><td>N_neigbours</td><td>5</td></tr>
<tr><td colspan="6">Software tools</td></tr>
<tr><td colspan="2">Google Colab</td><td colspan="2">Scikit-learn</td><td colspan="2">NLTK</td></tr>
<tr><td colspan="6">Dataset Split</td></tr>
<tr><td colspan="4">Training Set Size</td><td colspan="2">80%</td></tr>
<tr><td colspan="4">Testing Set Size</td><td colspan="2">20%</td></tr>
</table>

We got the dataset for this study from a public ML repository called the SMS Spam Collection. Table 4.1 displays the 5,574 SMS messages in this sample. Of those, 747 (13.4%) were classified as spam, while 4,827 (86.6%) were classified as valid or "ham" communications. We divided our SMS message sample among training and testing sections in order to evaluate the effectiveness of these algorithms. To be more precise, 80%

of the data were utilized to train the model, with the remaining 20% being saved for testing and assessment at the end. In machine learning, using 80/20 training-testing ratio is a popular practical method to guarantee reliable model performance evaluation.

4.4 Performance evaluation and discussion

The dataset utilized for the SMS spam identification challenge consisted of two classes: "Ham" (genuine communications) and "Spam". The confusion matrix, which showed the distribution of true and predicted classes, was used to evaluate the performance of the learned machine learning model: False Positives: Ham wrongly classed as spam, False Negatives: Spam wrongly categorized as ham, and True Positives: Spam accurately recognized. True Negatives: Ham was correctly recognized as non-spam. The researchers were able to calculate various performance metrics by analyzing the values in the confusion matrix, such as Accuracy-overall proportion of correct predictions, Precision-proportion of true positives among all Spam predictions, Recall-proportion of Spam messages correctly identified, and F1-Score.

Table: 4.2. Classification metrics

Class	Predicted	
	Ham	Spam
Ham	TN	FP
Spam	FN	TP

The success of the model may be inferred by computing a number of performance indicators through the analysis of the parameters in the confusion matrix. Here, the suggested method achieves an astounding 98% accuracy, highlighting the dependability

and potency of the created SMS spam detection technique. This section examines their dedication to providing outstanding SMS detection solutions as well as their in-depth knowledge of the issue domain. Below is a discussion of the thorough assessment performance and the experimental outcome in detail.

4.3 Classification performance for Multinomial Naïve Bayes classifier

Model Name	Accuracy	Precision	Recall	F1 Score
Multinomial Naïve Bayes	0.96	1.0	0.69	0.82

Table 4.3 presents an accuracy of 95.94% as well as a precision of 100%, the MNB classifier performs exceptionally well on test data, as demonstrated by the fact that all samples that were predicted as positive (class 0) were in fact positive. But with a recall score of only 69.57%, this model is not as good at accurately recognizing every positive sample—it overlooked 30.43% of the real positive samples. Additionally, the model's overall performance is measured by the F1-score of 82.05%, which strikes a reasonable balance between recall and accuracy.

Table 4.4. Classification performance for Support vector machine classifier

Model Name	Accuracy	Precision	Recall	F1 Score
SVM	0.98	0.97	0.86	0.91

The SVM classifier performed exceptionally well on the given classification test, as shown in Table 4.4. With a classification accuracy of 97.87%, the SVM model successfully categorized over 98% of the tested samples. With just three false positives and nineteen false negatives, the confusion matrix shows that the model produced very few errors. The model's accuracy score of 97.54% indicates that it can reliably identify positive samples, and its recall score of 86.23% indicates that it can also effectively locate the majority of real positive occurrences. With accuracy and recall taken into consideration, the model's total performance is balanced by its high F1-score of 91.54%. The SVM model performs better than the Multinomial Naive Bayes classifier that was previously mentioned with

regard to of accuracy, precision, recall, and F1-score, which makes it a good option for this classification task.

Table 4.5. Classification performance for Random Forest classifier

Model Name	**Accuracy**	**Precision**	**Recall**	**Precision**
Random Forest	0.97	0.99	0.81	0.89

Table 4.5 presents an accuracy rate of 97.39% on the test set, the RF classifier performs robustly on the given classification job. The confusion matrix reveals that the RF model made very few mistakes, with only 1 false positive and 26 false negative. The model's precision score of 99.12% is exceptional, indicating that it is highly accurate in identifying positive samples. When contrast to the SVM classifier, a recall value of 81.16% indicates that the algorithm missed a greater percentage of the real positive examples. A balanced indicator of the algorithm's overall performance that accounts for both accuracy and recall is the F1-score of 89.24%. The RF model performs somewhat worse than the SVM classifier in terms of accuracy, but it has better precision as well as lower recall. The SVM-based classifier may perform better overall than the RF model, as indicated by the RF model's F1-score being smaller than the SVM model's.

Table 4.6 Classification performance for AdaBoost classifier

Model	Accuracy (%)	Precision	Recall	F1 Score
AdaBoost	0.97	0.95	0.83	0.89

Table 4.6 shows that the AdaBoost model, with 100 estimators and a random state of 42, exhibited strong performance on the given classification task. Focusing on the accuracy metric, the AdaBoost model achieved an impressive accuracy score of 0.9729. This shows that 97.29% of each sample in the test set were properly classified by the model, indicating that it was highly effective in predicting the real class labels.

Examining the result of the confusion matrix, the AdaBoost model made 5 false positive predictions and 23 false negative predictions. Comparing the model against its true positive as well as true negative predictions, it appears that the model had a somewhat greater percentage of false negatives but a reasonably low proportion of false positives. Upon examining the accuracy result, the AdaBoost model attained a noteworthy 0.9583. This indicates that there was little to no false positives in the model's ability to correctly identify the positive cases. This is consistent with the data presented in the matrix of confusion. The AdaBoost model's recall score was 0.83; meaning that a sizable part of the test set's real positive events could be identified by the model. This implies that there was an appropriate level of false negatives in the model. In the end, the AdaBoost model's F1-score—which accounts for both accuracy and recall—was 0.8915. The model appears to have achieved a solid overall performance by maintaining a healthy balance between accuracy and recall, as indicated by the high F1-score. Overall, this suggested model performed remarkably well, showing excellent F1-score, accuracy, and precision. The model did reasonably well overall the classification challenge, despite the somewhat poorer recall suggesting that there is still space for improvement in detecting all the good examples.

4.7 . Classification performance for KNN classifier

Model	Accuracy	Precision	Recall	F1 Score
KNN	0.90	1.0	0.25	0.40

Table 4.7 shows The KNN model achieved an accuracy of 0.9003868471953579, or around 90.04%. This means the model correctly classified about 90% of the test samples with the confusion matrix values TP=35, FP=0, FN=103 and TN=896. The precision score is 1.0, which is excellent. This suggests that there had been no false positives and that the model was 100% correct when it predicted a positive class. However, the recall value is a pitiful 0.253. This indicates that only around 25% of the test set's real positive events were accurately detected by the model. A significant number of the good instances were overlooked. Simultaneously, the F1-score, which strikes a compromise between recall and

accuracy, is 0.404. Additionally, the model may have had difficulty striking a reasonable balance between recall and accuracy, as indicated by the relatively low F1-score.

In conclusion, the suggested KNN model showed good accuracy, properly classifying every positive prediction as a real positive. However, it was not very good at recalling the positive situations, missing a lot of them. This led to a low F1-score and an overall accuracy of about 90%. Despite its excellent accuracy, the model appears to have had difficulty fully identifying the positive class. The performance evaluation and overall results of the suggested method are displayed in figure 4.2. The accuracy, precision, recall, and F1-score performance metrics for each of the five suggested algorithms—M-NB KNN, SVM, Random Forest, and AB—are visually compared in this bar chart.

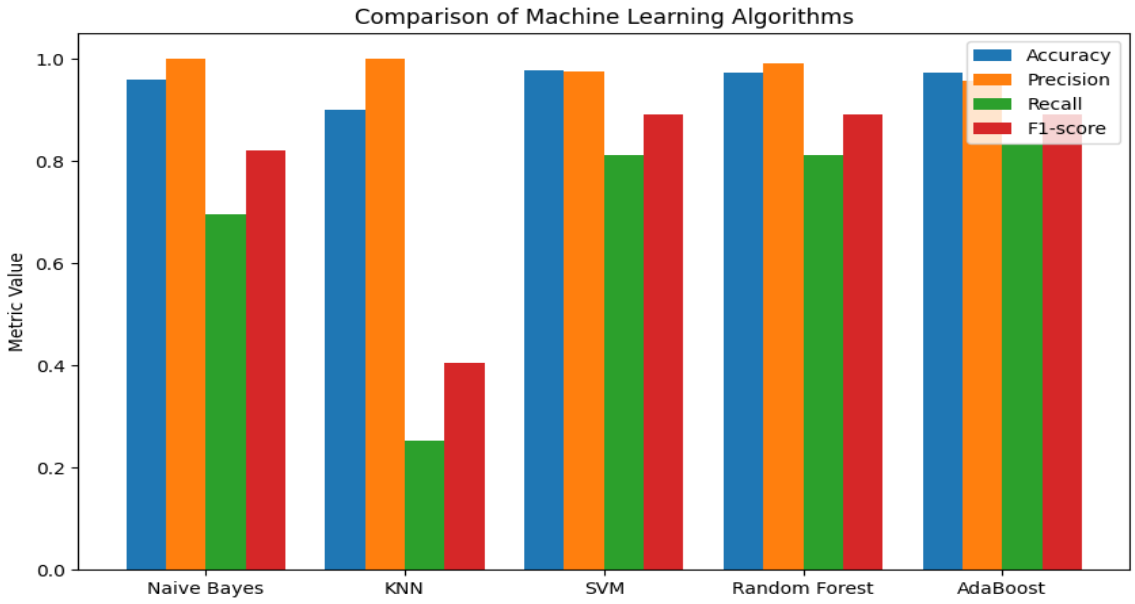

Figure 4.1 Performance comparison of classification algorithms

According to the comparison, the Random Forest and SVM classifiers perform the best across the board, although the SVM model surpassing the RF by a little margin. The KNN model has the weakest performance, while the AB model exhibits a mixed performance, with high recall but lower accuracy and precision. Let us look across each four key performance indicator metrics.

Accuracy:

- ✓ The classifiers with the best accuracy are the Random Forest and SVM models, with the SVM model marginally surpassing the Random Forest.
- ✓ When compared to SVM and Random Forest, the Multinomial-Naive Bayes and KNN models are less accurate.
- ✓ Out of the five methods, the KNN model is the least accurate.

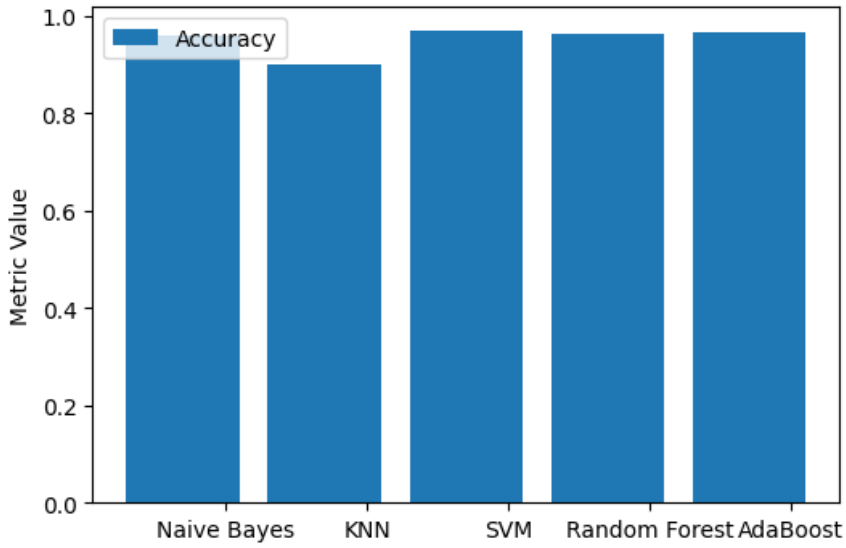

Figure 4.2 Comparison across accuracy

Precision:

- ✓ The SVM and Random Forest models have the highest precision scores, indicating their ability to accurately identify positive samples.
- ✓ The Naive Bayes and AdaBoost models have lower precision comparing to the KNN and Random Forest.
- ✓ SVM model has the lowest precision among the five algorithms.

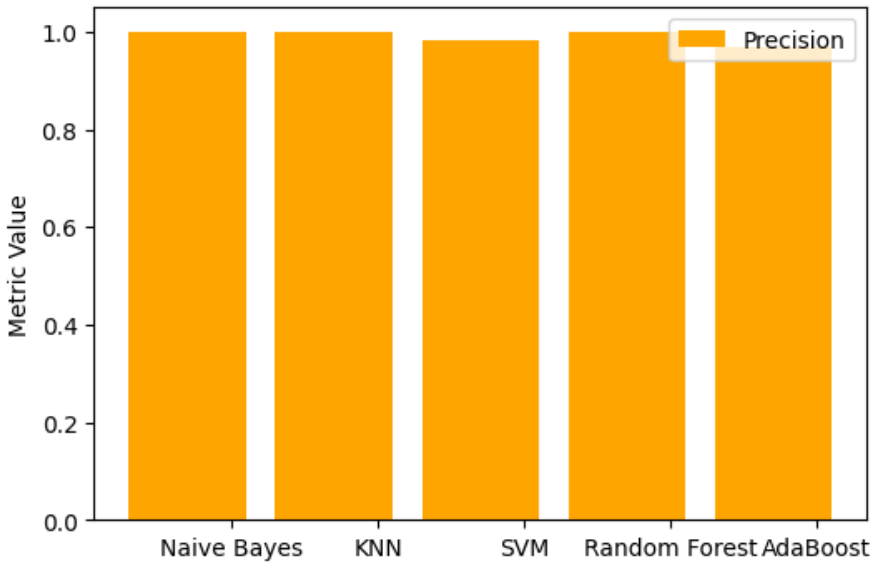

Figure 4.3 Comparisons across precision

Recall:

- ✓ The SVM and AdaBoost models have the highest recall scores, suggesting they are effective at finding most of the actual positive instances.
- ✓ The KNN model has a lower recall compared to the SVM and AdaBoost.
- ✓ Out of the five methods, the Naive Bayes & KNN algorithms have the lowest recall.

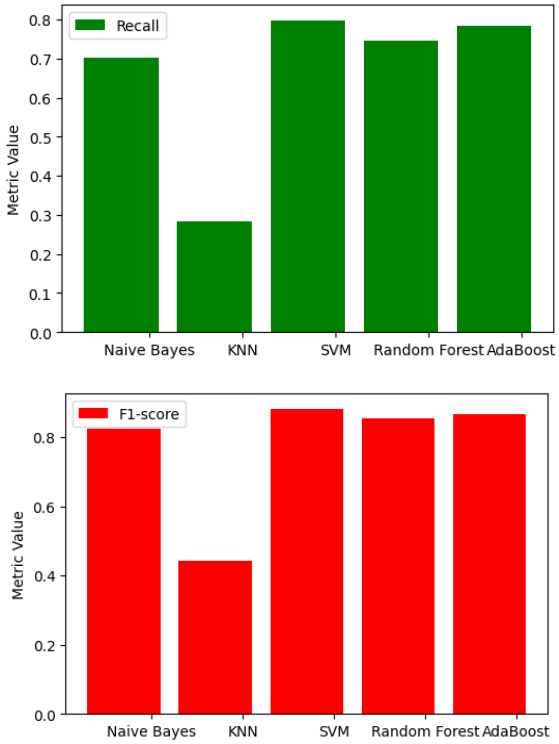

Figure 4.4 comparisons across recall (left) and F1_score (right)

F1-score:

- ✓ The model using SVM has the greatest F1-score, suggesting the optimal trade-off between recall and accuracy.
- ✓ Random Forest and AdaBoost models have slightly lower F1-scores compared to the SVM, but they still demonstrate strong overall performance.
- ✓ The KNN models have the lowest F1-scores among the five algorithms.

From the results we can say that the SVM and Random Forest are the most suitable algorithms for this classification problem.

Chapter 5 : Conclusion and Future Work

5.1 Conclusion

The SMS spam problem has become a serious and growing threat. It can cause seamless communication. This research work aimed to propose machine learning techniques which accurately and quickly detect SMS spam. The implementation and methodical assessment of machine learning techniques for precise SMS spam detection were covered. We obtained the text message spam dataset from a publicly accessible machine learning repository, and in order to better comprehend the data, we carried out extensive exploratory data analysis, followed by extensive preprocessing and cleaning steps - including removing duplicates, handling missing values, normalizing the text, and performing tokenization. We then leveraged Bag of Words and TF-IDF models for feature extraction, converting the unstructured message content into numerical representations, identify the most relevant attributes and optimize the model's performance and computational efficiency.

The simulation results show that SVM using a linear kernel performs better than other classifiers in this text messages spam detection assignment. As a consequence, our suggested method experiment result has attained an astounding 98% accuracy, surpassing the state-of-the-art traditional machine learning oriented spam detection approaches now in use for SMS spam detection. This high percentage of accuracy and effectiveness of the proposed strategy ensures that the system can rapidly identify and classify incoming messages as either legitimate or spam, without causing delays in the mobile communication process. By leveraging the capabilities of machine learning, this study has provided a significant advancement in the fight against the pervasive issue of SMS spam. The research outcomes enable the establishment of resilient and dynamic SMS spam detection systems, which are pivotal in fortifying communication channels and elevating the overall user experience.

5.2 Future Work

As the reliance on digital communication continues to grow, the threat of SMS spam has become a persistent challenge that demands robust and adaptive solutions. The work presented in this study has demonstrated the potential of classical machine learning techniques, such as M-NB, KNN, SVM, RF, and AB in tackling the SMS spam detection problem. However, to achieve more precise and reliable outcomes in the future, a holistic approach that incorporates additional informational indices should be considered. Extending the dataset used to train the model is an important subject for future research. By incorporating datasets from a variety of sources and ensuring a larger volume of records, the models can be exposed to a more diverse range of SMS patterns, leading to enhanced generalization capabilities. This approach will not only improve the overall accuracy of the models but also increase their reliability in real-world scenarios.

Furthermore, future work should focus on leveraging a comprehensive set of features beyond the traditional content-based characteristics. Non-semantic attributes that can shed light on the fundamental trends and contexts of spam messages include the SMSC initiator, Responses route, HTTP hyperlinks, Mobile stations Worldwide ISDN Number (MSISDN), as well as Protocol Identification codes like TP-PID in mobile text messages. By integrating these multi-dimensional features, the machine learning models can make more informed decisions, leading to a more robust and accurate SMS spam detection system.

Reference

[1] M. Gupta, A. Bakliwal, S. Agarwal, and P. Mehndiratta, “A Comparative Study of Spam SMS Detection Using Machine Learning Classifiers,” 2018 11th Int. Conf. Contemp. Comput. IC3 2018, Nov. 2018, doi: 10.1109/IC3.2018.8530469.

[2] D. Hintze, P. Hintze, R. D. Findling, and R. Mayrhofer, “A Large-Scale, Long-Term Analysis of Mobile Device Usage Characteristics,” Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol., vol. 1, no. 2, pp. 1–21, Jun. 2017, doi: 10.1145/3090078.

[3] S. Mishra, “Learning from Usage Analysis of Mobile Devices,” Procedia Comput. Sci., vol. 167, pp. 1648–1655, Jan. 2020, doi: 10.1016/J.PROCS.2020.03.375.

[4] Ericsson, “Ericsson Mobility Report (June 2019),” Ericsson White Pap., 2019.

[5] T. R. Graeff and S. Harmon, “Collecting and using personal data: Consumers’ awareness and concerns,” J. Consum. Mark., vol. 19, no. 4, pp. 302–318, 2002, doi: 10.1108/07363760210433627/FULL/XML.

[6] N. Islam and R. Want, “Smartphones: Past, present, and future,” IEEE Pervasive Comput., vol. 13, no. 4, pp. 89–92, Oct. 2014, doi: 10.1109/MPRV.2014.74.

[7] “Text messages sent in the U.S. 2021 | Statista.” Accessed: Jun. 11, 2024. [Online]. Available: https://www.statista.com/statistics/185879/number-of-text-messages-in-the-united-states-since-2005/

[8] “Social media penetration UK 2024 | Statista.” Accessed: Jun. 12, 2024. [Online]. Available: https://www.statista.com/statistics/507405/uk-active-social-media-and-mobile-social-media-users/

[9] R. Chaganti, B. Bhushan, A. Nayyar, and A. Mourade, “Recent trends in Social Engineering Scams and Case study of Gift Card Scam,” Oct. 2021, Accessed: Jun. 11, 2024. [Online]. Available: https://arxiv.org/abs/2110.06487v1

[10] M. Edwards, C. Peersman, A. Rashid, and S. Lancaster, “Scamming the scammers: Towards automatic detection of persuasion in advance fee frauds,” 26th Int. World Wide Web Conf. 2017, WWW 2017 Companion, pp. 1291–1299, 2017, doi: 10.1145/3041021.3053889.

[11] N. K. Nagwani and A. Sharaff, “SMS spam filtering and thread identification using bi-level text classification and clustering techniques,”

http://dx.doi.org/10.1177/0165551515616310, vol. 43, no. 1, pp. 75–87, Dec. 2015, doi: 10.1177/0165551515616310.

[12] T. M. Mahmoud, A. Mahfouz, and A. M. Mahfouz, “SMS Spam Filtering Technique Based on Artificial Immune System,” Artic. Int. J. Comput. Sci. Issues, 2012, Accessed: Jun. 11, 2024. [Online]. Available: www.IJCSI.org

[13] C. N. Lee, Y. R. Chen, and W. G. Tzeng, “An online subject-based spam filter using natural language features,” 2017 IEEE Conf. Dependable Secur. Comput., pp. 479–484, Oct. 2017, doi: 10.1109/DESEC.2017.8073830.

[14] H. Baaqeel and R. Zagrouba, “Hybrid SMS spam filtering system using machine learning techniques,” Proc. - 2020 21st Int. Arab Conf. Inf. Technol. ACIT 2020, Nov. 2020, doi: 10.1109/ACIT50332.2020.9300071.

[15] S. Gadde, A. Lakshmanarao, and S. Satyanarayana, “SMS Spam Detection using Machine Learning and Deep Learning Techniques,” 2021 7th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2021, pp. 358–362, Mar. 2021, doi: 10.1109/ICACCS51430.2021.9441783.

[16] D. V. R. D. J. S. S. Abiramasundari, “Spam filtering using Semantic and Rule Based model via supervised learning,” Ann. Rom. Soc. Cell Biol., pp. 3975–3992, Mar. 2021, Accessed: Jun. 11, 2024. [Online]. Available: http://annalsofrscb.ro/index.php/journal/article/view/1405

[17] S. J. Delany, M. Buckley, and D. Greene, “SMS spam filtering: Methods and data,” Expert Syst. Appl., vol. 39, no. 10, pp. 9899–9908, Aug. 2012, doi: 10.1016/J.ESWA.2012.02.053.

[18] A. Gayathri, J. Aswini, and A. Revathi, “Classification Of Spam Detection Using Naive Bayes Algorithm Over K-Nearest Neighbors Algorithm Based On Accuracy,” NVEO - Nat. VOLATILES Essent. OILS J. | NVEO, vol. 8, no. 5, pp. 8516–8530, Nov. 2021, Accessed: Jun. 11, 2024. [Online]. Available: https://www.nveo.org/index.php/journal/article/view/2247

[19] N. A. Patel and R. Patel, “A survey on fake review detection using machine learning techniques,” 2018 4th Int. Conf. Comput. Commun. Autom. ICCCA 2018, Dec. 2018, doi: 10.1109/CCAA.2018.8777594.

[20] “‘A third of TripAdvisor reviews are fake’ as cheats buy five stars.” Accessed: Jun.

12, 2024. [Online]. Available: https://www.thetimes.com/article/hotel-and-caf-cheats-are-caught-trying-to-buy-tripadvisor-stars-027fbcwc8

[21] S. Yu, “Factors influencing the use of mobile banking: the case of SMS-based mobile banking.” 2009. Accessed: Jun. 12, 2024. [Online]. Available: https://hdl.handle.net/10292/666

[22] “Mobile SMS Banking: Benefits, Use Cases & Security Concerns.” Accessed: Jun. 12, 2024. [Online]. Available: https://messente.com/blog/mobile-sms-banking

[23] A. K. Jain and B. B. Gupta, “Rule-Based Framework for Detection of Smishing Messages in Mobile Environment,” Procedia Comput. Sci., vol. 125, pp. 617–623, Jan. 2018, doi: 10.1016/J.PROCS.2017.12.079.

[24] T. Xia and X. Chen, “A discrete hidden Markov model for SMS spam detection,” Appl. Sci., vol. 10, no. 14, Jul. 2020, doi: 10.3390/APP10145011.

[25] P. Teja Nallamothu and M. Shais Khan, “Machine Learning for SPAM Detection,” Mar. 2023.

[26] S. DasGupta, S. Saha, and S. K. Das, “SMS Spam Detection Using Machine Learning,” J. Phys. Conf. Ser., vol. 1797, no. 1, p. 012017, Feb. 2021, doi: 10.1088/1742-6596/1797/1/012017.

[27] W. H. Gomaa, “The Impact of Deep Learning Techniques on SMS Spam Filtering,” IJACSA) Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 1, 2020, Accessed: Jun. 11, 2024. [Online]. Available: www.ijacsa.thesai.org

[28] L. Das, L. Ahuja, and A. Pandey, “A novel deep learning model-based optimization algorithm for text message spam detection,” J. Supercomput., pp. 1–26, May 2024, doi: 10.1007/S11227-024-06148-Z/FIGURES/9.

[29] N. Nur, A. Sjarif, Y. Yahya, S. Chuprat, H. Firdaus, and M. Azmi, “Support Vector Machine Algorithm for SMS Spam Classification in The Telecommunication Industry,” vol. 10, no. 2, 2020.

[30] U. Srinivasarao and A. Sharaff, “Machine intelligence based hybrid classifier for spam detection and sentiment analysis of SMS messages,” Multimed. Tools Appl., vol. 82, no. 20, pp. 31069–31099, Aug. 2023, doi: 10.1007/S11042-023-14641-5/TABLES/7.

[31] O. Abayomi-Alli, S. Misra, and A. Abayomi-Alli, “A deep learning method for

automatic SMS spam classification: Performance of learning algorithms on indigenous dataset," Concurr. Comput. Pract. Exp., vol. 34, no. 17, p. e6989, Aug. 2022, doi: 10.1002/CPE.6989.

[32] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text Classification Algorithms: A Survey," Inf. 2019, Vol. 10, Page 150, vol. 10, no. 4, p. 150, Apr. 2019, doi: 10.3390/INFO10040150.

[33] K. Chatzilygeroudis, I. Hatzilygeroudis, and I. Perikos, "Machine Learning Basics," Intell. Comput. Interact. Syst. Des., pp. 143–193, Feb. 2021, doi: 10.1145/3447404.3447414.

[34] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," SN Comput. Sci., vol. 2, no. 3, pp. 1–21, May 2021, doi: 10.1007/S42979-021-00592-X/FIGURES/11.

[35] Y. Baştanlar and M. Özuysal, "Introduction to Machine Learning," Methods Mol. Biol., vol. 1107, pp. 105–128, 2014, doi: 10.1007/978-1-62703-748-8_7.

[36] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," Electron. Mark., vol. 31, no. 3, pp. 685–695, Sep. 2021, doi: 10.1007/S12525-021-00475-2/TABLES/2.

[37] D. Jakhar and I. Kaur, "Artificial intelligence, machine learning and deep learning: definitions and differences," Clin. Exp. Dermatol., vol. 45, no. 1, pp. 131–132, Jan. 2020, doi: 10.1111/CED.14029.

[38] S. S. Mousavi, M. Schukat, and E. Howley, "Deep Reinforcement Learning: An Overview," in Lecture Notes in Networks and Systems, 2018. doi: 10.1007/978-3-319-56991-8_32.

[39] A. Q. Khan, M. Riaz, and A. Bilal, "Various Types of Antenna with Respect to their Applications : A Review," Int. J. Multidiscip. Sci. Eng., vol. 7, no. 3, pp. 1–8, 2016.

[40] N. K. Chauhan and K. Singh, "A review on conventional machine learning vs deep learning," 2018 Int. Conf. Comput. Power Commun. Technol. GUCON 2018, pp. 347–352, Mar. 2019, doi: 10.1109/GUCON.2018.8675097.

[41] "Applied Data Science Program | MIT Professional Education." Accessed: Jun. 12, 2024. [Online]. Available: https://professional-education-gl.mit.edu/mit-applied-data-science-

course?&utm_source=google&utm_medium=search&utm_campaign=ADSP_Gen_Course_UAE_Saudi_MEA&campaign_id=21242180810&adgroup_id=158976488102&ad_id=698042314268&utm_target=kwd-300255896069&Keyword=machine learning&placement=&gad_source=1&gclid=CjwKCAjw65-zBhBkEiwAjrqRMLBpHD9cXYgD2_8F8w6IR2pCY5QswzlDSF9pRF9NZil9piNcPRXnvhoCKyUQAvD_BwE

[42] "Logistic Regression for Machine Learning - MachineLearningMastery.com." Accessed: Jun. 12, 2024. [Online]. Available: https://machinelearningmastery.com/logistic-regression-for-machine-learning/

[43] K. M. Al-Aidaroos, A. Abu Bakar, and Z. Othman, "Naïve Bayes variants in classification learning," Proc. - 2010 Int. Conf. Inf. Retr. Knowl. Manag. Explor. Invis. World, CAMP'10, pp. 276–281, 2010, doi: 10.1109/INFRKM.2010.5466902.

[44] "K Nearest Neighbours — Introduction to Machine Learning Algorithms | by Sachinsoni | Medium." Accessed: Jun. 12, 2024. [Online]. Available: https://medium.com/@sachinsoni600517/k-nearest-neighbours-introduction-to-machine-learning-algorithms-9dbc9d9fb3b2

[45] "Support Vector Machines (SVM): An Intuitive Explanation | by Tasmay Pankaj Tibrewal | Low Code for Data Science | Medium." Accessed: Jun. 12, 2024. [Online]. Available: https://medium.com/low-code-for-advanced-data-science/support-vector-machines-svm-an-intuitive-explanation-b084d6238106

[46] "Random Forest. Random Forest is an ensemble machine… | by Deniz Gunay | Medium." Accessed: Jun. 11, 2024. [Online]. Available: https://medium.com/@denizgunay/random-forest-af5bde5d7e1e

[47] "Ensemble Learning — AdaBoost with Python | Tirendaz Academy | Medium." Accessed: Jun. 11, 2024. [Online]. Available: https://tirendazacademy.medium.com/ensemble-learning-adaboost-with-python-8332778fbb61