REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

USING OF MACHINE LEARNING FOR SPAM AND MALWARE DETECTION IN EMAILS BASED ON REGRESSION ALGORITHM

Rajaa Mohammed Hamzah AL-MAMOOR

Master's Thesis

Supervisor

Asst. Prof. Dr. Oğuz KARAN

İstanbul, 2022

USING OF MACHINE LEARNING FOR SPAM AND MALWARE DETECTION IN EMAILS BASED ON REGRESSION ALGORITHM

Rajaa Mohammed Hamzah AL-MAMOOR

Information Technologies

Master's Thesis

ALTINBAŞ UNIVERSITY

2022

The thesis titled USING OF MACHINE LEARNING FOR SPAM AND MALWARE DETECTION IN EMAILS BASED ON REGRESSION ALGORITHM prepared by RAJAA MOHAMMED HAMZAH AL-MAMOOR and submitted on 12/12/2022 has been **accepted unanimously** for the degree of Master of Science in Information Technologies.

Asst. Prof. Dr. Oğuz KARAN

the Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Oğuz KARAN	Department of Software Engineering, Altınbas University	_____
Asst. Prof. Dr. Sefer KURNAZ	Department of Computer Engineering, Altınbas University	_____
Assoc. Prof. Dr. Adil Deniz DURU	Department of Trainer Education, Marmara University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master's Thesis.

Submission date of the thesis to the Graduate Education Institute: ____/____/____

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Rajaa Mohammed Hamzah AL-MAMOOR

Signature

DEDICATION

I devote and pledge this research work to my supervisor who is salient for guiding me through whole research work as well as my family for always assisting me in my hard time.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisor Dr. Oguz KARAN for guiding and helping me along the way in writing this dissertation. Discussing my progress, problems, and ideas with my supervisor Dr. Oguz KARAN a couple of times every week helped me tremendously in understanding the logic behind the research. It made me better ealize the technical need for this research work.

ABSTRACT

USING OF MACHINE LEARNING FOR SPAM AND MALWARE DETECTION IN EMAILS BASED ON REGRESSION ALGORITHM

Al-Mamoor, Rajaa Mohammed Hamzah
M.Sc., Information Technologies, Altınbaş University,
Supervisor: Asst. Prof. Dr. Oğuz KARAN
Date: 12/2022
Pages: 61

Even though the size of these emails keeps growing, taking up more storage space on servers and requiring more bandwidth, people who receive emails spend a significant amount of time removing unwanted communications known as spam. This is even though these messages continue to grow in volume. Because of this, one of the most pressing issues of the present day is the development and enhancement of artificial classifiers that can differentiate between legal email and spam. When researching spam detectors, researchers have investigated various research approaches and feature sets to establish whether or not widely used terms were included in the spam. In commercial programming, there could be more than one approach used. Spammers are well aware of the efforts that are being taken to reduce spam, and as a result, they have developed a technique to bypass these filters.

On the other hand, human readers are adept at recognizing patterns in the evasive strategies spammers use. This study aimed to construct an alternative solution using a Logistic regression neural network classifier and train it on the dataset. The research was based on 5,573 spam emails that were sent by a variety of different people. A key step forward in this line of study is using descriptive qualities of words and messages analogous to those used by a human reader to identify spam as a foundation for picking the appropriate feature set. The information utilized while picking the feature forward served as the basis for the model used to choose the optimum feature set. As a bonus, we wanted to develop a neural network that

could reliably differentiate between spam and real email. Then we wanted to compare our findings to those obtained by other researchers.

Keywords: Spam Dataset, Legal Email, Commercial Programming, Logistic Regression, Evasive Strategies.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

HTTPS	:	Hyper Text Transfer Protocol Secure
URL	:	Uniform Resource Locator
SGO	:	Social Group Optimization
ANN	:	Artificial Neural Network
DNN	:	Deep Neural Network
KNN	:	K-Nearest Neighbors
SOM	:	Self-Organizing Map
SVM	:	Support Vector Machine
R.F	:	Random Forest
LSTM	:	Long-term/Short-term Memory
UDP	:	User Datagram Protocol
DNS	:	Domain Name System
DDoS	:	Distributed Denial-of-Service
ISPs	:	Internet Service Providers
NN	:	Neural Network
ESP	:	Email Service Provider
CC	:	Carbon Copy
BCC	:	Blind Carbon Copy
AI	:	Artificial Intelligent
SMTP	:	Simple Mail Transfer Protocol
GLM	:	Generalized Linear Model

1. INTRODUCTION

1.1 OVERVIEW

The term "spam" describes undesired information in the computer world. Instant Messenger (IM) and the telephone Short Message Service (SMS) have also lately been used to refer to unwanted messages delivered over email and Usenet (SMS). Spam, or unwanted, unsolicited email, always promotes acquiring a product or service [1].

Unsolicited Bulk Email (UBE) and Unsolicited Commercial Email (UCE) are other terms for spam-type email (UCE) [2]. In most circumstances, spam promotes online purchases; however, it may also drive sales over the phone or via other means on rare occasions.

A spammer is a person or organization that sends out unsolicited electronic communications, often known as spam; figure (1.1) appears spam-relaying countries. Spammers will employ a broad range of automated tools and techniques to deliver spam on behalf of companies, such as web crawlers. In many cases, spammers are the owners of their online businesses, which they advertise via spam emails [3].

Even though the content may be objectionable, emails from well-known senders are often not considered "spam". Consider the example of a never-ending list of jokes that different friends and acquaintances submitted. Many people don't think of viruses or Trojan horses as spam since they share some of their features with spam [4]; even though they are hazardous software. In the spam-fighting community, emails that don't qualify as spam are referred to as ham. If a message is deemed spam by one person, it may be welcomed by another. There is no widely recognized definition of spam.

In computer science, machine learning models have been used for a wide range of purposes, including eliminating problems related to network traffic and detecting viruses. A significant number of people now connect and socialize regularly via email. When customer data is exposed, spammers may utilize email addresses that have been hacked to send unwanted emails to customers (spam). In addition, phishing assaults may be used to get unauthorized

access to a user's device by persuading the user to click on the spam link contained in the spam email [5].

Companies provide a wide array of tools and approaches that may be used to identify spam emails that are sent over a network. Establishing rules and properly configuring the firewall settings have enabled organizations to recognize unsolicited emails [6].

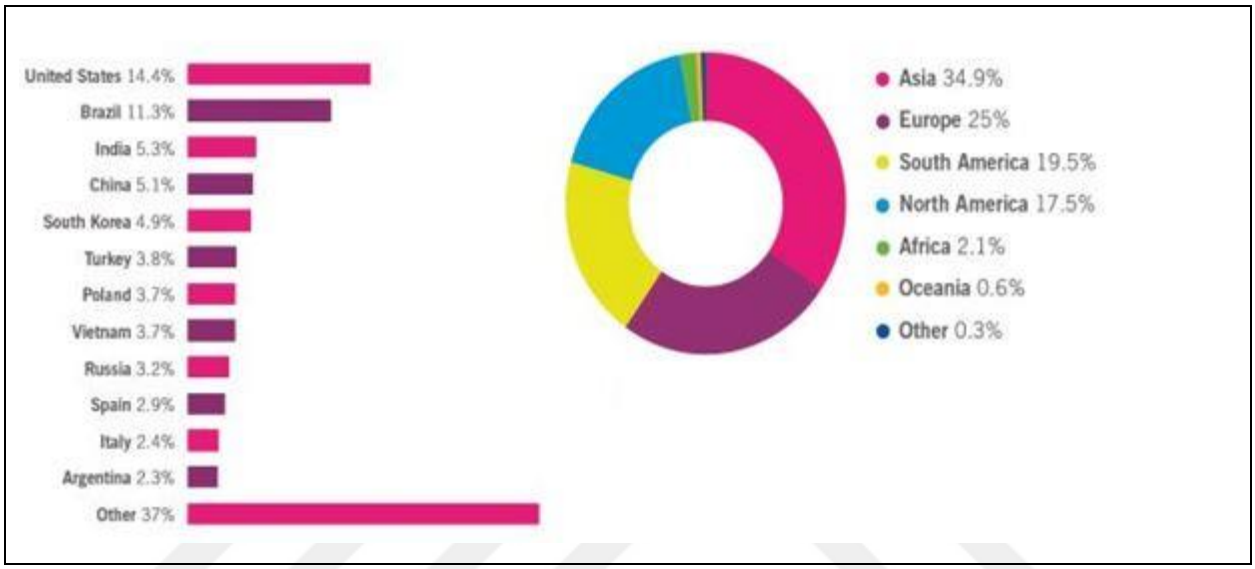

Figure 1.1: Spam by countries[3].

1.1.1 Filtering of Spam

The data is filtered in optimal form, as shown in figure (1.2). Individual messages should be transmitted one after the other. The learning-based filter assesses whether or not a message is a spam. Following that, a human evaluates the message to assess whether or not the anticipated label is correct and then provides feedback to the learning-based filter by informing it of this information [1], [7-8]. The filter may then utilize the label input to modify its internal model, ideally resulting in improved future predictions. This hypothetical best-case scenario is helpful for three reasons. This model provides a decent first approximation of the spam filtering environment in real-world circumstances. In these cases, messages come streaming, and users contribute label input. For starters, it is a natural extension of the online learning scenario, which has been widely researched in machine learning [8]. It's a natural match for the online-learning environment for three reasons. Various machine learning algorithms today might be used to complete this assignment. Third, this case demonstrates that there is no limit to the

quantity of information that may be provided. An urgent need is for updated and efficient systems adapting to shifting data patterns.

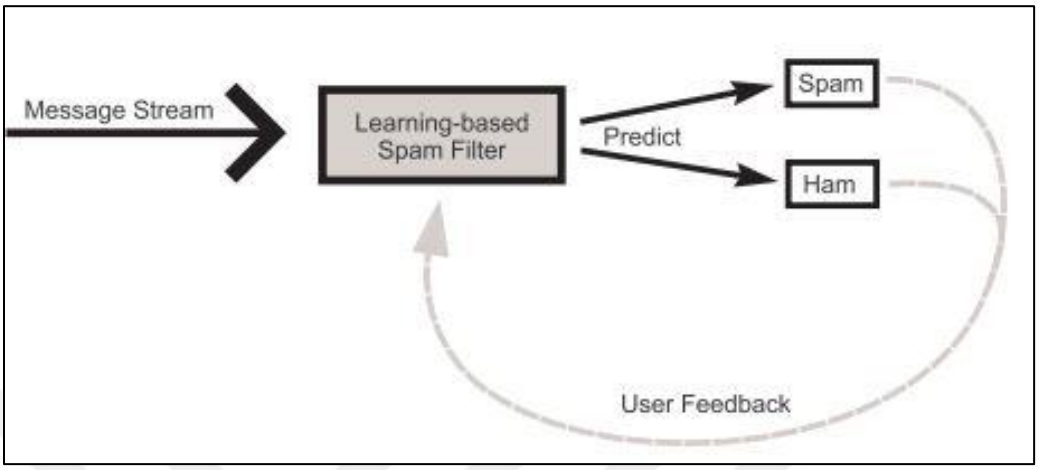

Figure 1.2: An idealized situation for web-based filtering.

1.2 MOTIVATION

Electronic mail, or email, is one of the most frequently used services on the internet because of its high transmission speed, capacity to handle multimedia content, and low distribution cost. According to figure (1.3), an average of 306.4 billion emails will be sent daily by 2020. By 2024, this number is expected to rise to 361.6 billion [9]. While more people are using email these days, there has also been an increase in unsolicited or unwanted emails. Sending an unsolicited commercial email, sometimes known as spam, poses a significant risk to both people and companies. Because of the amount of data it transmits, spam lowers the available bandwidth of networks and consumes more space in users' mailboxes. This makes the rapid dissemination of misleading information and the proliferation of malicious software more likely to occur. It's estimated that spam accounts for around 55% of all email traffic.

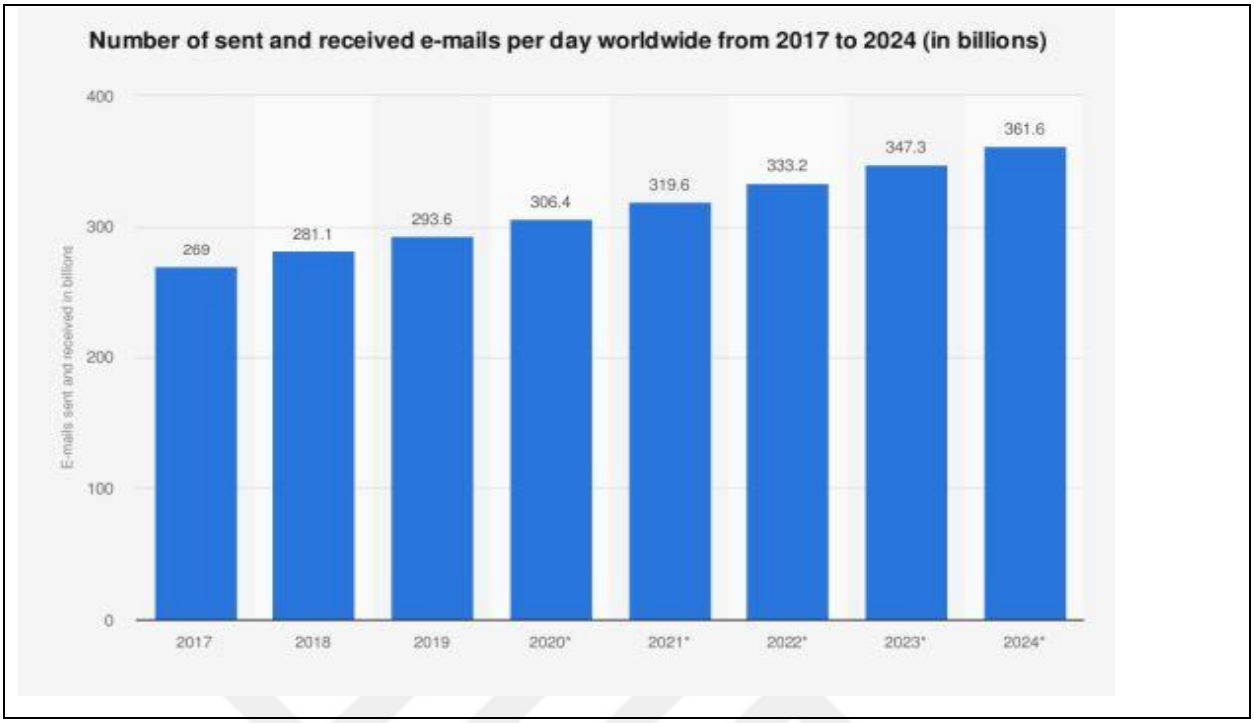

Figure 1.3: Daily email volume, 2017-2024 [9].

There are billions of dollars in lost productivity because of spam, which is at the base of the problem [1]. Using Microsoft Research and Google data, researchers found that spam costs U.S. businesses and consumers $20 billion annually, while spammers and spam-advertised merchants make $200 million annually [10].

1.3 PROBLEM STATEMENT

As the second quarter of 2017 closed, researchers found an increasing number of new phishing websites [11]. According to trend research conducted at the time, 190,942 phishing emails were discovered in the third quarter of 2017. Despite a decline in the number of sites found, there were 273,395 and 296,208 phishing complaints in the second and third quarters, respectively. Only four of the 54,631 distinct websites found used HTTPS out of the total [11].

Our study employed machine learning methods such as logistic Regression to find spam emails. Data sets were used, including real (important) and fake (spam) emails. Passwords, social security numbers, account numbers, and other sensitive information are often the targets of these assaults. When a target receives an email from an attacker, the victim clicks on a

phishing URL, which directs them to a fake website where they are asked to submit personal information [12].

Data might have come from reputable organizations like trade associations, banking institutions, and other businesses. The attackers first gather the victims' email addresses before launching their assaults. After that, the I.P. addresses are used to target fake accounts. As these spam communications proliferated quickly, immediate action was required to reduce the threat. For this method, both strength and speed are necessary.

Using blacklists, whitelists, signatures, and message header verification is a non-machine learning approach [13]. However, many approaches are both costly and time-consuming. In research, content-based approaches are often used. We need to classify emails more quickly and rigorously to spot phishing messages among the billions of incoming emails. The process must be finished fast to analyze the categorization technique's outcomes.

As a result, our study focused on figuring out the best remedies for this issue. For the screening methods we've found, Logistic Regression has shown to be a highly successful and unique categorization strategy. This method is known as "supervised learning" by researchers.

1.4 THESIS GOAL

The following are some of the goals of the current research:

a. Developing an efficient method for identifying spam reduces all types of assaults carried out through suspicious messages.
b. Analyzing the phrases used in phishing attack vectors and using the logistic regression classifier to detect the texts used in the assaults
c. Analyzing and contrasting the results of the proposal classifier on our dataset.
d. We used comparisons of under- and oversampling and the sampling technique to examine the results of our dataset.

1.5 CONTRIBUTION

This letter includes the following contributions from us:

a. Logistic Regression is used to find spam using a large open dataset called a "spam dataset," which has been evaluated (for object and topic) and sampled into several data sets of different sizes, each with the same number of spam and clean emails.

b. On test datasets, vector encoding of email text is applied in various ways.
c. It is then applied to the vector representations of the email text in each test data set using the proposed technique. After then, the data is analyzed, and the findings are summarised in this thesis.

1.6 SCOPE OF THE THESIS

This thesis applied techniques such as information retrieval and supervised machine learning. It was accomplished using a clustering approach that utilized word sorting to discover traits. The strategy presented was solely intended for use with text-based phishing assaults. Because no large and diverse datasets are widely available to the general public, a trained spam dataset was used for training and testing. We obtained data for our research through various tactics, including phishing emails and SMS. We obtained this dataset when we looked at this social media message.

2. LITERATURE REVIEW

This section provides a summary of some of the significant research that has been conducted on data mining approaches and algorithms to recognize phishing emails. There are just a few research initiatives now being conducted that are entirely focused on finding a solution to the problem of phishing attempts. Emails used for phishing are often associated with spam, and most of these strategies target spam management to prevent identity theft fraud. The most important difference is that phishing emails have appropriate feature selections, but spam emails do not. This is how spam is differentiated from phishing emails.

2.1 DETECTING MALWARE IN EMAIL

Research studies such as [14-15]; have been conducted to understand better the behavior of malware as a persistent cyber issue. The majority of viruses launch attacks and interact with their command and control servers via the use of network connections [16]. To achieve this objective, various techniques and processes have been devised to detect it via monitoring and analyzing the network and using certain characteristics of the network traffic. Methods to collect [17]; analyze [18]; classify, and detect [18-19]; the scientific community has actively presented malware.

Email spam ordinarily includes particular spontaneous messages sent in mass by individuals you do not know. The term spam is obtained from the Monty Python sketch [20]; in which the Hormel canned meat item has numerous tedious emphases. While the term spam was first utilized in 1978 to allude to unwanted email, it increased rapidly in the mid-1990s, as we became progressively typical outside scholastic and research circles [21]. increased rapidly in the mid-1990s, as we became progressively typical outside scholastic and research circles [21]. A notable model is the development expense trick, in which a client receives an email with an offer that should bring about a prize. In the era of technology, the dodger/spammer shows a story where the unfortunate casualty needs forthright financial help so that the fraudster can gain a lot bigger total of cash, which they would then share. The fraudster will either earn a profit or avoid communication when the unfortunate victim completes the installment [22],[23],[24].

2.1.1 Anomaly Detection

A pattern of behavior characterized as an anomaly deviates significantly from what would normally be predicted. Outliers and novelties are both terms that are used to refer to anomalies. Both terms are used to describe occurrences that are uncommon throughout a dataset. In contrast to outliers, novelties are the outcome of the normal model for the data changing over a period, leading to the emergence of new instances that are not consistent with the norm [25]-26].

Using backpropagation and backpropagation with momentum, S. Sinha et al. [27]; have devised a method to detect spam. The authors employed SGO (Social Group Optimization) to increase the model's classification accuracy. Backpropagation needs additional iterations, which increases the computation time.

Cidon et al. [28]; studied employee impersonation and business email compromise (BEC), and they developed a system called BEC-Guard to detect both. BEC and staff impersonation are more difficult to combat than conventional spam email filtering. This kind of assault is difficult to detect since it does not use malware and is tailored to the victim.

Karim et al. [29]; investigated A.I. strategies to identify spam emails. The most prevalent classifiers were ANN, DNN, Naive Bayes, Decision Tree, Random Forest, Logistic Regression, SVM, Adaboost, and KNN. In a modest number of research initiatives, hybrid approaches were used. K-Means and the Self-Organizing Map are two common unsupervised approaches (SOM).

Goodall et al. [30]; conducted a poll to get a deeper grasp of the NIDS user group and their views. We conducted semi-structured interviews with IDS specialists to better understand their methodologies. IDS tasks need a mix of broad and specialized knowledge (such as network security) (e.g. knowledge of normal network behavior).

According to Bahnsen et al. [31]; there are two approaches to locating phishing URLs on websites. Utilizing lexical and statistical URL analysis in conjunction with Random Forest (R.F.) classifiers was one approach to feature engineering that might be used. According to the authors, the model training accuracy of the second approach, which was the long-term/short-term memory (LSTM) neural network, was 0.98. In contrast, the model training accuracy of the first method, R.F., was 0.93.

Laorden et al. [32]; recently did some research that changed how anomaly detection filters spam emails. They found that it wasn't as important to put spam emails in different categories and that it was enough just to show one type of email. The review has three parts: a demonstration of the first anomaly-based spam sieving method, an improvement of the method that used a data minimization technique to the characterized dataset corpus to reduce processing time while keeping recognition rates the same, and an investigation into whether it is better to use spam or non-spam emails to show normality.

2.1.2 Statistical Detection

It has been shown that n-gram analysis can detect file segments' anomalies using File print analysis [33]; a statistical binary content analysis method for identifying malicious material hidden in files. This was achieved by parsing the binary content of files using the File print method. Using 1g and 2g analysis makes it possible to distinguish between typical system files and malware. Bi-grams represent uncompressed code, which differs from other methods because there is no valuable concern if the code is malicious or not. PAYL [31]; and n-gram [33]; are network intrusion detection systems that use n-gram analysis to identify anomaly packets.

According to the findings of AV/TEST [34]; the number of malware infections skyrocketed from 100 million in 2008 to 600 million in 2017 during those ten years. Websites and Internet of Things devices that are connected to the internet have the potential to exhibit harmful behavior.

According to CertN [35]; there is a possibility that users will get unauthorized access to websites. Various malware or dictionary-based software, such as brute force, may be used to access usernames, passwords, and any other login information.

Pandey and Saini [36]; researched attacks using TCP, I.P., and UDP protocols. According to what they claimed, network security should be safeguarded rather than just detected. Many approaches should be used to identify a network's vulnerability based on the three characteristics mentioned earlier. As a result, the current study provided machine learning techniques to recognize linked variables such as TCP, I.P., trash (email), and port and their potentially harmful behavior to predict future malware trends.

Li et al. [37]; provide a summary of DNS-based attacks and the findings of various other botnet studies at the paper's conclusion. The particular characteristics of a botnet, such as which are more likely to result in security flaws, were not the subject of the research. To achieve this goal, the present research analyzed data from botnets and identified crucial characteristics that point to the presence of malicious software.

Al-Garadi et al. [38]; DDoS, SQL injection, cross-site scripting (XSS), HTTPS token attacks, and web trace attacks are used in many malicious attacks.

2.2 EMAIL SPAM FILTERING

Emails that are undesired, unsolicited, or infested with viruses are prevented from entering email inboxes by spam filters. Internet service providers (ISPs) use spam filters to stop the spread of unwanted electronic messages known as spam. Large businesses are not the only ones that use spam filters to protect their workforces and networks from being bombarded with unsolicited emails [39-40].

As is standard practice in most businesses, both incoming and outgoing email traffic is screened by anti-spam software (email leaving the network) [39],[41]. Both of these approaches are used by Internet service providers (ISPs) as part of their commitment to the security of their customers. Small firms often place a focus on inbound filtering systems.

According to the results of Jindal and Liu [42]; the first effort to detect product review spam was conducted by analyzing the similarity between reviews and the qualities of the items. This was done to determine whether or not a review was spam. In particular, spammers' propensity to write several evaluations of identical products was used to one's benefit.

As a consequence of this early endeavor, researchers moved on to construct approaches for identifying review spam based on the cosine similarity of reviews, as stated by Lim et al. [43]. As a direct consequence of using this methodology, Liu et al. [44]; proposed using a probabilistic graph classifier founded on a bidirectional N.N. with an attention mechanism.

Sanz, Hidalgo, and Perez [45]; investigated how users are affected by email spam and how users and providers might lessen the impact of its negative effects. This paper explains several legal, economic, and technological techniques that might be used to counteract spam emails. Because content analysis filters enjoy widespread use, not to mention a reasonable degree of accuracy and precision, the evaluation concentrated on determining how these tools function.

S. Dhanaraj et al. [46] presented a concise study summary on several methods for filtering email image spam. This study's primary emphasis was developing anti-spam email filtering algorithms that were image-based rather than text-based. The purposeful decrease in spam and the spam filters put into place to fight it have increased the amount of invention and creativity achieved. Although the study did not consider machine learning methods, modeling tools, datasets, or the architecture of email spam filtering systems, the findings were nonetheless interesting.

Bhowmick and Hazarika [47] analyzed a few of the most commonly used content-based spam filtering methods. The research focused mostly on machine learning-based solutions for spam filtering. This research examined the major ideas, efforts, and outcomes of spam filtering to determine what was working and what was not. The fundamentals of email spam filtering, the dynamic nature of spam, and how spammers circumvent email service provider (ESP) spam filters were discussed. In addition, they analyzed the most prevalent machine learning algorithms used in spam prevention.

2.3 MACHINE LEARNING-BASED SPAM DETECTION

Due to the widespread interest in the subject, a diverse spectrum of scientific studies is included in the literature. Eye trackers [46],[48] have been used to understand how individuals interact with phishing emails and websites. [49–50] have attempted to take a more preventative stance by posing victims on phishing websites and providing bogus information. [51] looked at the aesthetic similarities and contrasts between legitimate websites and the phishing sites that imitate them. One-time passwords are another security measure analyzed concerning the battle against phishing attacks. Phishing URLs allowed have been the subject of research conducted by [52], while blocklisting tactics have been the focus of research conducted by [53].

Machine Learning Algorithms have been more popular among researchers in recent years (MLAs). [54] conducted research on phishing URLs using Logistic Regression to detect and find bogus URLs. This study was published.

Zareapoor et al. [55] discovered two different sorts of phishing attacks: one was called deceptive phishing, and the other was called malware phishing. As a direct consequence of this, they were successful in recognizing phony phishing efforts.

They characterized the procedure of accomplishing things as a four-step process. T SVM, Ada Boost, and Random Forest have each produced outcomes with more than 99.5% success rates. In addition to analyzing the content of emails, Fette et al. [56] looked at whether or not emails include JavaScript, the amount of dots and reference links they contain, and the total number of dots. So far, they have sent out 860 emails that are phishing scams and 6950 emails that are not phishing scams in English.

Toolan et al. [57] conducted their research by sending 4,116 phishing emails and 4,202 ham emails to participants. This experiment looked at many different algorithms, including C5.0, a decision tree method, k-NN, SVM, Nave Bayes, and linear Regression. SVM and C5.0 are two machine learning algorithms with the greatest accuracy rates compared to other machine learning algorithms. Researchers have shown an increased interest in machine learning algorithms in recent years (MLAs).

Even though the spam filters already in place are exceedingly efficient, Gallo et al. [58] claim that they do not identify all potential dangers. The researchers created a variety of machine learning strategies and put them to the test using a dataset consisting of around 12,000 emails that were either malicious or safe. Among them are Decision Trees, Linear Support Vector Machines, RBF Support Vector Machines, MLP Neural Networks, and Random Forests. According to the study's findings, there was no significant difference in performance between training with just eight characteristics and training with the whole set.

Cidon et al. [59] researched employee impersonation and corporate email compromise (BEC). As a result, a supervised approach known as BEC-Guard was developed to recognize both types of attacks. Screening for spam emails regularly is easier than dealing with BEC and employee impersonation, which are both more challenging. Because it does not include malware and is adapted to the person it is directed at; this attack is difficult to detect. The impersonation classifier uses various characteristics, including sender, recipient, CC, and BCC. In addition to that, the classifier takes advantage of Random Forest.

Karim et al. [60] researched several methods of detecting spam emails using A.I. assistance. It was discovered that the most popular classifiers include ANN, DNN, Naive Bayes, Decision Tree, Random Forest, SVM, Adaboost, and KNN. Logistic Regression was determined to be the least popular classifier. In a very modest number of studies, hybrid approaches were shown to be effective. K-Means and the Self-Organizing Map are two unsupervised approaches used rather often (SOM).

According to the authors, few studies use the I.P. source, destination information or the SMTP envelop information. According to research on recent developments in machine learning, two of the most used algorithms for supervised learning are the Naive Bayes algorithm and the Support Vector Machine (SVM) method.

Nandhini et al. [61] constructed a bagging-based machine learning model to detect spam. Two computer algorithms were used. J48 and Naive Bayes are algorithms (Decision Tree). The dataset is split and sent to the proper algorithms to detect spam using this strategy. This report's authors conducted three experiments. First, Naive Bayes is applied. Probability-based classifiers analyze data to determine the likelihood of a given category. The second experiment used J48 Decision Tree. Entropy is used to use training data to create decision trees. In the third trial, SMD identifies spam emails using the J48 method, the Naive Bayes Multinomial classifier, and a hybrid bagged approach.

Luo GuangJun et al. [62] assessed the ability of prominent machine learning algorithms to recognize spam emails. Classification techniques such as Bayesian classification, K-nearest neighbor classifier, neural network classifier, SVM classifier, artificial immune system classification, and rough sets classification may be found on this site. This section also contains a preliminary classification of sets. When something has occurred in the past, it is possible to estimate that it will occur again in the future. This technique is known as the Naive Bayes classifier method. After that, the email will be designated as spam or junk mail, depending on the likelihood that it includes spam or junk mail.

a. Regression

Regression is supervised learning that uses a set of characteristics to guess what a future value (Y) will be (X). In the regression model, continuous or quantitative variables are used. Regression is a simple method for machine learning that makes accurate predictions with little error. A representation of regression may be found in Equation (2.1).

$$Y = f(x) + E \tag{2.1}$$

Where Y is the followed (or output) mutable, x is the free (or input) mutable, f is a function that shows the relation between x and Y, and E is a random error that could happen.

3. LOGISTIC REGRESSION

3.1 BACKGROUND

Logistic Regression is an algorithm for Machine Learning used for classification problems. It is a predictive analytic technique that is based on the idea of probability. Some instances of classification issues are whether or not an email is spam, whether or not a transaction completed online constitutes fraud, and whether or not a tumor is malignant. To calculate and return a probability value, logistic regression must first modify its output using the logistic sigmoid function [63-64]. The logistic regression graph is depicted in figure (3.1).

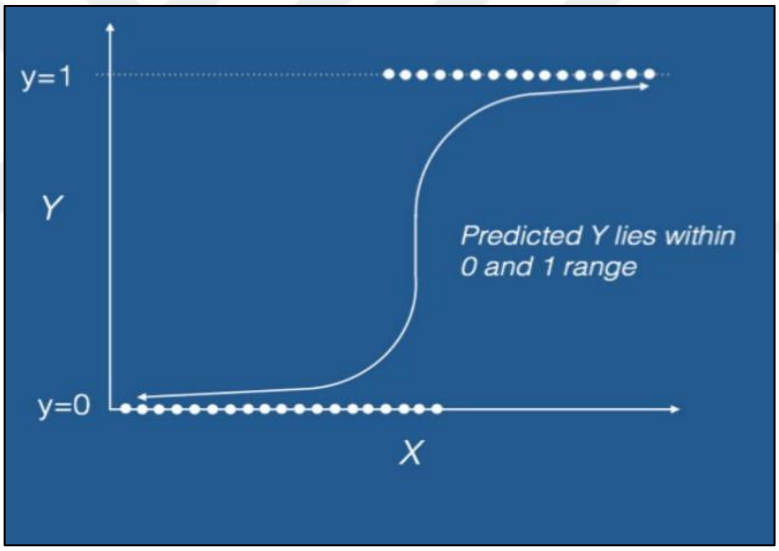

Figure 3.1: Logistic regression graph.

It is a statistical method that can indicate whether or not a binary answer variable, such as Y, is dependent on one or more independent variables, such as X =. (X1,....., Xn). It is a tool for developing a model in circumstances where there is a two-level categorical response variable, as opposed to scenarios in which there is a numerical response variable. Multiple linear regression would be a more appropriate analysis technique in this case. Like multiple regression, logistic regression is a generalized linear model (GLM3) type, with the categorical response variable serving as the difference [65-66].

In general, the result of a GLM is symbolized by the symbol Yi, where I refer to the observation number in question.

The value of Yi in this report will show whether or not an email is spam; (Yi = 1) will indicate that it is spam, and (Yi = 0) will indicate that it is not spam. The format used to represent the independent variables, represented by X, is as follows: The value of the variable j for the ith observation is represented by the symbol xij.

When the likelihood exceeds π i, the result Yi has the value (Yi = 1), and when it is less than (1- π i), it has the value (Yi = 0).

The logistic regression model makes use of a framework that is quite similar to that of multiple regression. This structure builds a link between the prediction factors and the likelihood that an email is spam πi. (x1i,.....,xij). Since the response is binary, we must find a suitable transformation for the regression model to work correctly. The symbol πi undergoes the logit transformation naturally, as demonstrated by the following example:

$$\text{logit}(\pi\,\text{i}) = ln\,(= \frac{\pi\,\text{i}}{1-\pi\,\text{i}}) \tag{3.1}$$

The formulas for the logistic regression model are as follows:

$$Ln = (\frac{\pi\,\text{i}}{1-\pi\,\text{i}}) = \alpha + \beta 1\text{xi}1 + \beta 2\text{xi}2 + \cdots \ldots \beta\text{jxij} \tag{3.2}$$

It's crucial to remember that while the log $(\frac{\pi\,\text{i}}{1-\pi\,\text{i}})$ might be any actual integer, the possibility that an email is spam πi could be any value between zero and one:

$$0 \leq \pi\, i \leq 1 \quad \longrightarrow \; -\infty < In\,(\frac{\pi\,\text{i}}{1-\pi\,\text{i}}) < +\infty \tag{3.3}$$

The relationship P(Yi = 1) can be found by substituting 2 for $\pi\, i$ in the equation. We get:

$$\text{P(Y = 1|X = } x\text{)} = \pi\, i = \frac{\exp(\alpha+\beta 1\text{xi}1+\beta 2\text{xi}2+\cdots\ldots\beta\text{jxij})}{1+\exp(\alpha+\beta 1\text{xi}1+\beta 2\text{xi}2+\cdots\ldots\beta\text{jxij})} = \frac{\exp(\alpha+\beta\text{x})}{1+\exp(\alpha+\beta\text{x})} \tag{3.4}$$

This investigation will be conducted using the logistic regression model, represented by equation (3.4); the odds might be defined as

$$\Omega = \frac{\pi i}{1 - \pi i} \tag{3.5}$$

The probability of the outcome being nothing but no spam is divided by the probability that the event will be spam. Equation 2 is reached after applying the logarithmic transformation to both sides. The logistic regression coefficients, calculated independently for each variable, are equivalent to the change in the log odds. The exponentiated version of the coefficients can indicate the odds ratio [67].

3.2 MECHANISM OF LOGISTIC REGRESSION

Consider organizing your email as an example. We will parse the emails for information pertinent to the query, such as the email's sender, to evaluate whether or not an email is a spam. There are several errors in the email. Using expressions or words like "offer," "prize," "free present," etc.

An array of numerical properties is created from the previously stated information. A score between 0 and 1 is generated due to the logistic function's transformation and the linear combination of numerical features. This rating shows the possibility that an email is spam rather than official correspondence. The email will be classified as spam if the probability is higher than fifty percent. Figure (3.2) displays the comparison between ar linear regression and logistic regression

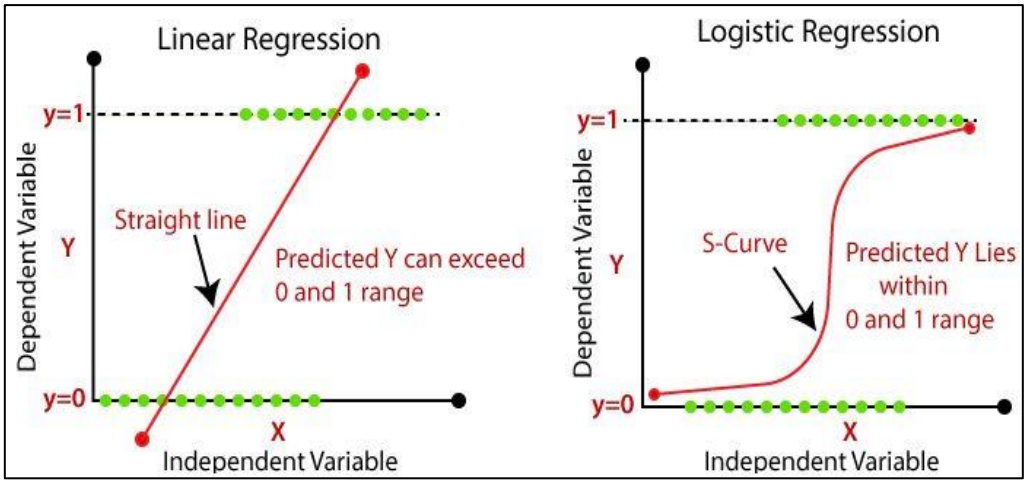

Figure 3.2: Linear regression vs logistic regression.

Two of the most well-known examples of machine learning algorithms that fall within the category of supervised learning techniques are linear regression and logistic regression. Since both of these algorithms are supervised in nature, to produce their predictions, these algorithms utilize datasets that have been labeled. How each one is put to use, though, is where the primary distinction lies. Linear Regression is utilized for the problem-solving process of Regression, whereas Logistic Regression is utilized for the problem-solving process of Classification.

When solving classification issues, we are tasked with estimating the likelihood that a given outcome variable falls into a specific category. When linear regression is used for classification, the classes or categories that are being analyzed will be represented as numbers. It will conform to the line that provides the greatest fit by minimizing the distance that separates the data points and the line. Simply giving scores along the best fit line is all that the linear regression equation would do [68]. These ratings are not to be construed as possibilities in any way. Specifying a reasonable threshold that will differentiate the classes is impossible.

In addition, the linear regression model corresponds to a straight line that can be extrapolated. It is possible for values to fall outside of their normal range, either below 0 or above 1 (-∞ to ∞). In logistic regression, a logistic function is applied so that the dependent variable can only take values between 0 and 1; this is necessary because the probability lies within a limited range between 0 and 1.

3.3 LOGISTIC FUNCTION

It is also called the sigmoid function, is an S-shaped curve that is capable of transforming any real-valued integer into a number between 0 and 1 by utilizing the equation as follows:

$$f(x)= \frac{1}{1+e^{-x}} \tag{3.6}$$

In the illustration in figure (3.3), f(x) changes from 0 to 1 as the variable x gets closer to the symbol and changes from 1 to 0 as it gets closer to the symbol∞.

$$f(x)= \frac{1}{1+e^{-\infty}} = 1 , \quad e^{-\infty} \longrightarrow 0 \tag{3.7}$$

$$f(x)= \frac{1}{1+e^{-(-\infty)}} = \frac{1}{1+e^{\infty}} = 0 , \quad 1/ \infty \longrightarrow 0 \tag{3.8}$$

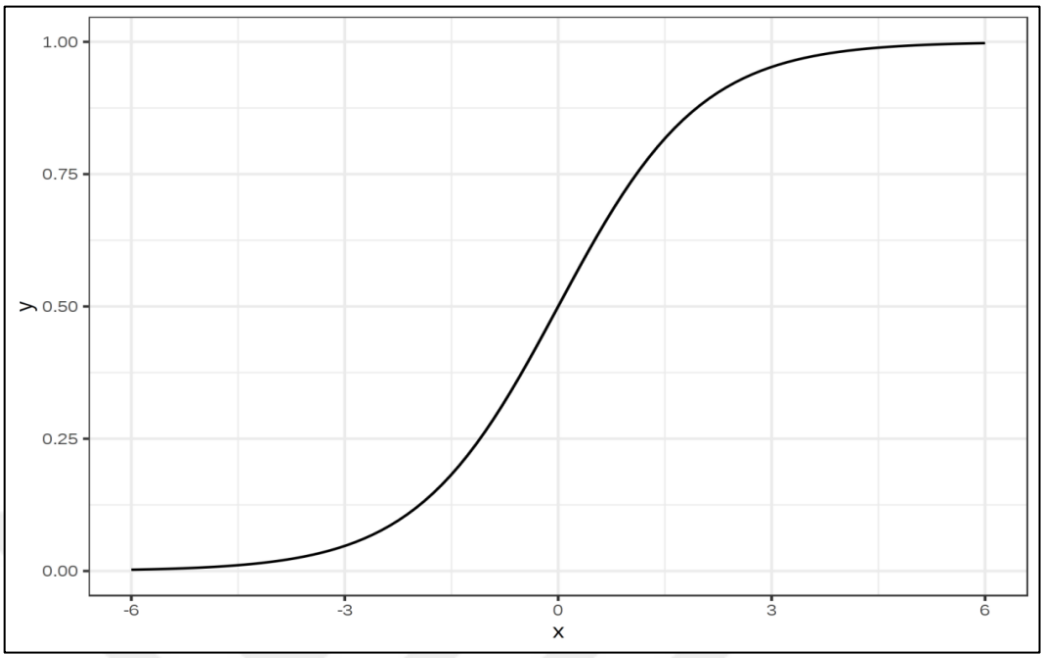

Figure 3.3: Logistic function.

4. METHODOLOGY

In this chapter, we will talk about the frameworks, the preparation of the dataset, the system's execution, and the algorithm utilized, and describe the numerous stages that make up the methodology.

4.1 TECHNIQUE CLASSIFICATION CONCEPT

One of the methods by which the machine learns is through classification. Its primary objective is to correctly categorize the missing parameters of the attribute of the targeted recognized values [69]. Classification is essential to data mining and machine learning since it provides a definite demarcation between the different classes. This is accomplished by developing an in-depth comprehension of the relationship between the parameters and the class characteristic [70-71]. It has been established that some characteristics differ in a minor way or that the distinction between them is negligible. Because of this, results can be acquired in the shortest time possible if some less important traits are disregarded. The Logistic Regression classifier was utilized in the development of a model that we created to categorize breast cancer. The Ham-Spam dataset was collected from the Kaggle site archives and used for training and testing the model.

4.2 DESIGNING SOFTWARE

Jupyter Notebook is an environment for writing code in Python. It comes equipped with a computational intelligence framework Sci-Kit Learn, used to build the model suggested. In addition to having constructed support for all computer learning algorithms currently in use for categorization, Sci-Kit Learn also includes a good variety of tools for data pre-processing techniques and computational performance measurements. This language offers important benefits over others due to its adaptability, which allows for easy plotting of graphical form and the provision of output following the converge phase of education.

4.3 PROPOSED SYSTEM

The recommended model for this research is called logistic regression, and we decided to use it to train on the dataset we were given. Data gathering, a preprocessing step that deals with

the treatment of missing data, training and testing of machine learning models, and success assessment and comparison are the modules that comprise the suggested framework. The proposed architecture is depicted in Figure (4.1), and the data were obtained from the online repository on the Kaggle site. The information gathered will be preprocessed, the pretreatment will manage any missing values in the data, and a characteristic scaling technique will be used to normalize the data. The preparation will take place before the data is normalized.

A training set, a testing set, and a validation set are each constructed from the collected data. While the test set is utilized for the goal of validation, the training set is used for training the prediction model. The predictive model is analyzed and contrasted using these performance measures, which include accuracy, precision, recall, and specificity, as well as the f1 score.

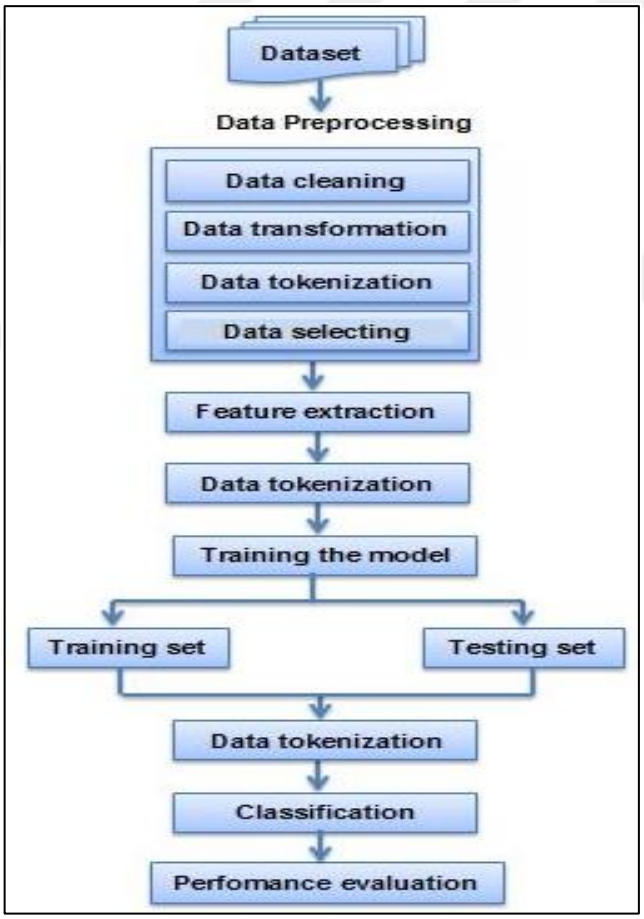

Figure 4.1: Methodology for workflow design proposed.

4.4 EXPERIMENTS

The first stage in the technique is the pre-processing of the data, which is accomplished with the help of the tools found in the SciKit Learn package, which is accessible through the computer language Python (version 3.6.5).

Considering the dataset that will be utilized, the pre-processing will center on managing the missing attributes, the unbalanced data, and the number of attributes that will be used to train the prediction model.

To address the missing values, we calculated the mean of the values that were not missing. This mean value will replace any missing values for any characteristics in the dataset with one or two missing values.

In investigating issues involving unbalanced data, it is essential to alter either the classifier or the training set balance, or even both, to prevent the development of an erroneous classifier. When dealing with datasets that are not evenly distributed, it is usual to practice rebalancing them artificially, a process that is referred to as "up-sampling" (replicating characteristics from the minority) and "down-sampling" (removing cases from the majority). Several research shows that this method does not significantly impact the prediction performance of learned classifiers [72], and these studies can be found all over the internet.

Using methods of machine learning that are immune to the effects of unbalanced data is the approach used in this study to overcome the problem of uneven data distribution. Logistic Regression is the name of this classifier.

4.5 DATA PREPARATION

When comparing the efficacy of various spam filters, it is common to practice using multiple benchmark datasets. Selecting datasets from those areas to demonstrate that the suggested spam filtering model is relevant across various application domains is essential. It is essential to evaluate how effectively the suggested model functions across various data contexts, including the degree to which data is sparse and the degree to which classes are imbalanced.

The Spam-Ham dataset is a well-known personalized dataset that contains email messages that are both spam and ham. As seen in the table (4.1), the spam dataset utilized in various research has a total of 5573 emails, with 83% genuine emails and 17% spam emails. The messages are transmitted in their initial form; nevertheless, they are encoded using a non-Latin script and have undergone a few minor adjustments (legitimate emails sent by the owners of the mailboxes to themselves and a handful of virus-infected emails are removed). Every message is stored in its text document.

Table 4.1: Distribution of data on ham and spam.

Type of message	Percentage	Ham	No. of data
Ham	83%	4825	5573
Spam	17%	747	

The database of SMS texts has 5 columns when viewed in pandas: v1 (which classifies each text or email as spam or ham), v2 (that includes the text contents themself), and three Unidentified columns. The titles of the v1 and v2 columns will be changed to class labels and messages accordingly, while the other columns are eliminated. Figure (4.2) displays the data used and appears parts of messages in the dataset.

	class_label	message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
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 aro...
...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...
5568	ham	Will Ì_ b going to esplanade fr home?
5569	ham	Pity, * was in mood for that. So...any other s...
5570	ham	The guy did some bitching but I acted like i'd...
5571	ham	Rofl. Its true to its name

5572 rows × 2 columns

Figure 4.2: Screenshot of exploring the dataset.

Characteristics of Ham-Spam Dataset:

a. To study SMS Spam, we've compiled a large set of messages labeled as such; these messages constitute the SMS Spam Collection. It is a single collection of 5,573 English-language SMS messages that have been classified as either "ham" (legal) or "spam."
b. Each line in the files contains a single message. Each line has two columns: v1 has the label (either ham or spam), and v2 is where the actual content is stored.
c. This corpus was constructed using free or freely available research online sources.
d. The Grumbletext website was manually combed through, and a collection of 425 SMS spam messages was obtained there. This is a forum in the United Kingdom where users of mobile phones make public accusations about spam text messages, the vast majority of which are made without reporting the actual spam message received. It was an extremely challenging and time-consuming operation that included meticulously examining hundreds of web pages to identify the text of spam messages contained inside the claims.
e. The NUS SMS Corpus (NSC) is a dataset of approximately 10,000 legitimate messages collected for research at the Department of Computer Science at the National University of Singapore. This particular subset consists of 3,375 SMS messages that were chosen at random and are considered to be ham messages. The communications were primarily sent by people living in Singapore, most of whom were students at a local university. Volunteers, all of whom were informed that their contributions would be made available to the general public, provided the input used to compile these messages.
f. A compilation of 450 SMS ham messages taken from Caroline Tag's PhD research.
g. We have included the SMS Spam Corpus version 0.1 Big in our database. It is open to the public and contains 1,002 SMS ham messages and 322 spam messages.
h. There was a reoccurring instance of a particular phrase or character in the emails.
i. The run-length attributes (52-58) are used to determine the total length of sequences that consist entirely of capital letters.

4.6 DATA PREPROCESSING

Preparing the data is one of the essential and initial steps that must be taken during data mining before it can be fed into an algorithm. This is a crucial action.

It is a technique for organizing a dataset from raw data. The data must be formatted, sanitized, and sampled as part of this procedure.

4.6.1 Data Cleaning

A dataset can be "cleaned" by deleting any inaccurate, damaged, badly formatted, redundant, or missing data. This process is referred to as "data cleaning." There is a higher chance that part of the data will be duplicated or erroneously classified when several data sources are integrated. Only the most recent and crucial files and documents are preserved thanks to the data cleansing process, making it simpler to find them whenever needed. Additionally, it helps ensure that not a lot of personally identifying information, which could be a security risk, is kept on the computer.

4.6.2 Normalization

Deep learning routinely uses normalization, a data preprocessing method that can be considered an essential step. The goal of the method is to convert the values of numerical columns in a dataset to a similar scale while maintaining the value ranges. This is necessary since each feature is measured on a unique scale and contains a unique range of possible values. When dealing with a scenario like this, it is typically beneficial to homogenize the data to scale each characteristic to a broad range [73]. In this study, the challenge of model learning was overcome by ensuring that the various features had value ranges that were comparable to one another. This allowed gradient descents to converge at a faster rate.

4.6.3 Data Transformation

To ensure that the algorithm will work properly and accept legitimate input, it is important to adjust the data in question, considering both the algorithm that will be applied to them and the data that will be worked on. Three of the most common transformations that can be applied to the data include scaling, deconstructing, and aggregating. When the attributes in question are numerical, some algorithms are sensitive to the range of possible values for each characteristic. The data in these circumstances must be normalized. It may be difficult for the algorithms to use some of the data in their current form since they include complex ideas. These data will be much more useful once separated into their constituent parts. For instance, the bag-of-words model can be used to vectorize the text data in an email,

and the algorithm can then use this vectorized data. The bag-of-words model will eventually be explained in greater detail. A feature more applicable to an algorithm may occasionally be created by combining many characteristics into one feature.

4.6.4 Data Tokenization

The process of breaking down a long string of characters into smaller units, such as words, phrases, sentences, or other parts, is known as data tokenization. These little things are referred to as tokens. As examples of common heuristics used for this, the sequence can be interrupted on white spaces, punctuation letters, or newline characters. White spaces and punctuation marks are removed from the text as part of the procedure. A variety of libraries and tools can be used to do this procedure. The NLTK library, made available in Python, is used for tokenizing this thesis [74-76].

4.6.5 Data Selecting

Email message fragments are condensed into feature vectors using a feature selection technique, reducing the geographic coverage. When a large message necessitates a compact feature representation, this method is useful for quick text or image matching [77]. Typical advance fee fraud schemes include inheritance, lotteries, visas, customs clearance, etc.
Sender Account Features utilized for spam filtering include the Sender's Country, IP Address, Email, Age of Sender and Recipient, and Sender Reputation. Less significant pieces of information are the sender's birthday, password, account age, sender's sex, and recipient's age. Detecting spam emails using the fewest available attributes is crucial due to the computational workload and time required. Stemming, reducing noise, and removing stop words are a few of the feature selection process's stages [78].

4.7 IMPLEMENTATION OF SYSTEM

4.7.1 Importing the Data

In this stage, we will perform preliminary processing and preparation of the data to use it in our code effectively. Figure (4.3) displays the code that should be used for this:

```
#Data Pre-procesing Step
# importing libraries
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
#importing datasets
data_set= pd.read_csv('user_data.csv')
```

Figure 4.3: Screenshot of code for data preprocessing.

4.7.2 Label Encoding

Python's label encoding requires that the category value be replaced with a numeric value that falls between 0 and the number of classes, which is 1. Figure (4.4) displays the code that should be used for this:

```
# label spam mail as 0;  ham mail as 1;

mail_data.loc[mail_data['Category'] == 'spam', 'Category',] = 0
mail_data.loc[mail_data['Category'] == 'ham', 'Category',] = 1
```

Figure 4.4: Screenshot of code for label encoding.

4.7.3 Data Sampling

Several sampling strategies may be required depending on the quantity and quality of the data and the available computational and memory resources. For instance, one possibility is to examine a more manageable representative subset of the data if the amount of data that is accessible is far higher than what is needed or if the available resources cannot process such a vast amount of data. Additionally, when the data is skewed, such as when the dataset has a big number of spam emails but a small number of nonspam emails, spam emails can be sampled to match the number of nonspam emails.

4.7.4 Splitting the Data

The data can be separated into training and testing sets in several ways. The most prevalent method is employing a random sampling method in one of its many forms. Implementing a technique that relies solely on randomness from start to finish is not only easy, but it also helps to ensure that the process is not prejudiced in any way towards any aspect of the data. Figure (4.5) displays the code that should be used for this:

```
# Splitting the dataset into training and test set.
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=3)
```

Figure 4.5: Screenshot of code for data splitting.

4.7.5 Feature Scaling

This step of the pre-processing of the data is concerned with independent variables and data attributes. This support essentially increases the comparability of the data within a given range. It can also be useful in speeding up the computations that an algorithm performs. Since we want our predictions to be as accurate as possible, we shall scale the features in logistic regression. We will only scale the independent variable in this case because the dependent variable can only have values of 0 and 1. Figure (4.6) displays the code that should be used for this:

```
#feature Scaling
from sklearn.preprocessing import StandardScaler
st_x= StandardScaler()
x_train= st_x.fit_transform(x_train)
x_test= st_x.transform(x_test)
```

Figure 4.6: Screenshot of code for feature scaling.

4.7.6 Feature Extraction

The purpose of feature extraction is to reduce the number of features in a dataset by generating new features based on the ones already present (and then discarding the original features). After trimming down, this new collection of features should be capable of summarising most of the information in the initial set of features. Extracting features from the input data is the first step in improving the learned models' accuracy. By eliminating superfluous or duplicate information, this stage of the general framework brings the dimensionality of the data down to a lower level. Naturally, both the learning and inference times are shortened as a result. Figure (4.7) displays the code that should be used for this:

```
# transform the text data to feature vectors that can be used as input to the Logistic re
gression

feature_extraction = TfidfVectorizer(min_df = 1, stop_words='english', lowercase='T
rue')

X_train_features = feature_extraction.fit_transform(X_train)
X_test_features = feature_extraction.transform(X_test)

# convert Y_train and Y_test values as integers

Y_train = Y_train.astype('int')
Y_test = Y_test.astype('int')
```

Figure 4.7: Screenshot of code for feature extraction.

4.7.7 Training the Model

All of the experiments on the classifiers mentioned in this research were carried out using a machine learning package called SciKit Learn, and Jupyter Notebook was used.

SciKit-Learn is a built-in library in Python 3.6.5 that contains a variety of machine learning techniques for data pre-processing, categorization, regression, clustering, and classification methods. These machine learning techniques may be found in SciKit-Learn.

The methods of machine learning included in SciKit-Learn are used to solve a wide variety of problems that occur in the real world. Experimenters and software developers have access to a well-defined framework within the program, which allows them to construct and test their models.

The factors in Binary Logistic Regression analytical techniques are dummy variables, which use various size levels.

On the other hand, the factors considered have to be linear and satisfy the response necessary for this method. After performing a non-linear conversion on the linear regression model, one obtains a logistic regression model as the final product.

Once the dataset has been cleaned and preprocessed, we may divide it into training and testing data using the "train test split" function. To deploy the training data on the model and assess whether the questioned text is spam, performance metrics and the Logistic Regression algorithm from the "scikit-learn" library must be imported. In our project, we used the Logistic Regression technique we created to solve the classification task. With the potent predictive modeling technique of logistic regression, classification problems with two or more possible outcomes can be predicted with probability. Similar to linear regression, logistic regression yields an S-shaped line as opposed to the output of linear regression, which is a straight line with 0s and 1s. Logistic regression uses the sigmoid function to generate an S-shaped curve. The sigmoid function can be used to calculate probabilities in the range of 0 and 1. Logistic regression will inform us whether the message is spam in the context of our model. If the value is one, it is regarded as spam; if it is zero, it is regarded as ham. Figure (4.8) displays the code that should be used for this:

```
model = LogisticRegression()

# training the Logistic Regression model with the training data
model.fit(X_train_features, Y_train)
```

Figure 4.8: Screenshot of code for training the model.

Let us say you check your phone and find the following message: ""CONGRATULATIONS!! Your email address was selected at random to receive a lottery prize of $2,500,000.00 USD. To collect your reward, please contact our office at the following addresses: Addin55@gmail.com.hk or +62 024 118 3370. "The terms "lottery," "prize," "office," and "email" all come up here.

W = [0.3, 0.3, 0.1, 0] exp *T* is the equation for the weight vector that has been provided.

The following factors will determine whether or not the email is spam:

x=[1,1,1,2] exp T

wTx=0.3∗1+0.3∗1−0.1∗1−0.04∗2=0.42>0

$$\Pr(y = 1|x) = \sigma(w^T x) = \frac{1}{1+e^{-0.42}} = 0.603 \quad (4.1)$$

Figure (4.9) provides a graphical representation of the Testing and Training Model that can be seen

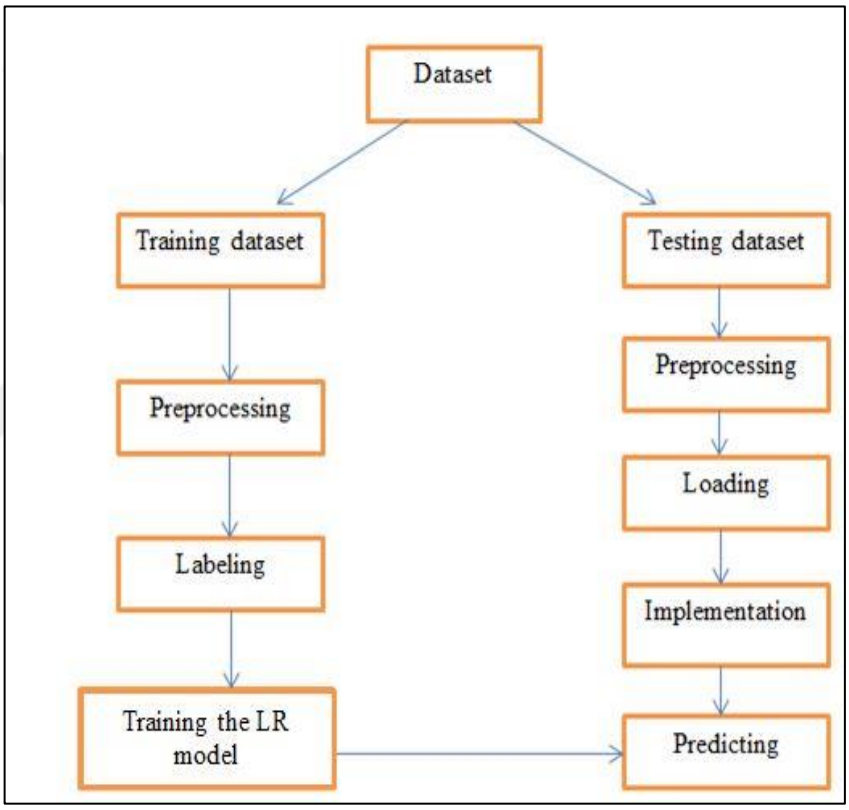

Figure 4.9: Testing and training model flowchart.

4.7.8 Evaluating the Trained Model

Building statistical formulas are an integral part of machine learning, which is used to comprehend the data better. After these models have been "fitted" to past data, they may be used to make predictions based on data that has yet to be viewed. We can assess the model performance with the assistance of various model evaluation approaches, enabling us to compare multiple models that were fitted to the same dataset. In addition to assessing how well the model performs on our training data, we also do it on our test dataset,

which contains data that has not previously been seen. Figure (4.10) displays the code that should be used for this:

```
# prediction on training data

prediction_on_training_data = model.predict(X_train_features)
accuracy_on_training_data = accuracy_score(Y_train, prediction_on_training_data)


# prediction on test data

prediction_on_test_data = model.predict(X_test_features)
accuracy_on_test_data = accuracy_score(Y_test, prediction_on_test_data)
```

Figure 4.10: Screenshot of code for predictive performance of the trained model.

An accuracy score, which represents the model's overall precision, is one method that can be used to evaluate a logistic regression model. We might use other measures such as accuracy, Recall, confusion matrix, and so on to examine how well the classifier performs within a particular class (positively and negatively predictive ability). To make improvements to the prediction model, we can choose from a variety of available approaches:

a. The preliminary processing of data. The most significant advancements are typically accomplished through an effective data cleaning procedure.
b. The scaling of features. Values of features can vary significantly from one another by several orders of magnitude.
c. Regularization.

4.8 PERFORMANCE MEASURES

Machine learning models that are trained using datasets require several parameters to get an accurate and unambiguous assessment of their performance. Because these concepts were investigated during our work, we decided to use them as the primary standards to evaluate participants' performance in this study.

4.8.1 Accuracy

Compares the appropriate proportion of cases that have been detected by making an educated guess about the total number of messages [79].

$$accuracy = \frac{\text{true detected cases}}{\text{all cases}} \tag{4.2}$$

$$ACC = \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}} \tag{4.3}$$

4.8.2 Sensitivity (Recall)

Machine learning is a metric used to evaluate the effectiveness of a model in identifying positive examples. It's often referred to as the TPR or recalls. When gauging a model's efficacy, we look at its sensitivity to see how many positive examples it accurately identified [80].

$$SE = \frac{\text{TP}}{\text{TP+FN}} \tag{4.4}$$

4.8.3 Specificity

It is the statistic that determines how well a model can predict true negatives for each of the categories that are given [81].

$$SP = \frac{\text{TN}}{\text{TN+FP}} \tag{4.5}$$

4.8.4 Precision

It is determined by the ratio of the total number of samples classified as positive to the number of positive samples identified as positive [81].

$$Pre = \frac{\text{TP}}{\text{TP+FP}} \tag{4.6}$$

4.8.5 F1-Score

It neatly summarises a model's predictive performance by combining its rival measurements, recall, and accuracy.

$$FI - score = 2\ x\ \frac{Pre\ x\ SE}{Pre+SE}\ +\ \frac{2\ x\ TP}{2\ x\ TP+FP+FN} \tag{4.7}$$

4.8.6 Reciever Operating Curve (ROC)

(ROC) stands for receiver operating characteristic and is used to measure the efficacy of solutions to classification issues with two classes. It's a curve of probabilities showing how the signal is isolated from the background noise as the TPR is plotted against the FPR at different cutoff points.

5. RESULTS

The preprocessing, training and validation phases of the prediction models are covered in this chapter, along with the outcomes of the constructed framework and the programming language used throughout those phases. To support our proposed framework, screen captures of the outcomes are shown here.

5.1 PRESENTATION RESULTS

All the phases of preprocessing, normalization, testing and training of data, categorization, and accuracy assessments are accomplished using the SciKit Learn collection in Python programming. These steps include managing missing values of a variable, feature scalability of the data, training the predictive model, and assessing the models' results in terms of accuracy, precision, and sensitivity. In addition, this research work presents all of the steps that were taken during this research work. All of the results are presented and discussed in detail.

We utilized the Logistic Model Tree algorithm to classify the data and the 10-fold cross-validation test to evaluate it; the latter is a technique for assessing prediction models that uses a subset of the full dataset for training and the remaining subset for testing. First, we checked each message's header, keywords, and whitelist/blacklist to extract feature vectors for use in training the datasets. The trained models' performance in terms of classification accuracy is assessed using 10-fold cross-validation.

One measure of email spam classification performance is the rate at which new spam is identified and classified. It's quantified by dividing the number of successfully classified examples by the training dataset's total number of test cases. False negatives occur when spam emails are incorrectly labeled as non-spam and delivered to the user's inbox. When an email is falsely identified as spam, it is either deleted or relegated to the spam folder. Mistaking legitimate messages for spam can be much more expensive for users than actually getting spam. One of the measures used to measure an email spam filter's efficiency is its false positive rate.

5.2 FILE READING

The data obtained from the Kaggle site repository can be found on my local system in the directory "C:\Users\MODE\Desktop\MY THESIS-NEW \CODING," and the file that contains the data is called content/drive/MyDrive/mail data.csv. This directory can be accessed by clicking here. In the Python programming language, a library known as Pandas can be used to open and read comma-separated values (CSV) files. The Python code used to read our data set file is seen in Figure (5.1).

Figure 5.1: Screenshot of reading in the Raw_mail_data file.

5.3 DATA PRE-PROCESsING

To deal with the missing, we decided to implement a technique in which we would fill in the missing values by using the estimated mean of the values in the dataset that were not missing. To implement this, we used an imputer method available in SciKit Learn to deal with the missing value scenario. The python code we used to deal with the missing values is displayed in Figure (5.2) below. If we don't deal with the missing values, it will be difficult to train the classifiers.

```
[ ]  # replace the null values with a null string
     mail_data = raw_mail_data.where((pd.notnull(raw_mail_data)),'')

[ ]  # printing the first 5 rows of the dataframe
     mail_data.head()
```

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
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 aro...

```
[ ]  # checking the number of rows and columns in the dataframe
     mail_data.shape

     (5572, 2)
```

Figure 5.2: Screenshot of filling the null value in the ham-spam dataset.

5.4 EVALUATION OF CLASSIFIER TRAINING EXPERIMENTS

The preprocessed data are put through a logistic regression model before being used for categorization. The data is divided into three parts: the training set, which comprises 75% of the total data; the testing set, which comprises 20% of the total data; and the validation set, which comprises 5% of the total data. When we wanted to evaluate how well the model was performing, we utilized the test set, but when we wanted to train it, we used it. The Python programming code utilized to do this is displayed in Fig. 4.6. There are 4180 data points in the training set, 1115 data points in the validation set, and 278 data points in the testing set. The distribution of the dataset used to create sets is shown in table (5.1).

Table 5.1: Training, testing, and validation data distribution.

Dataset	No. of data	Training set	Testing set	Validation set
Ham-Spam	5573	4180	1115	278

The Python code that was utilized to put the training of the Logistic Regression model into action is displayed in Figure (5.3).

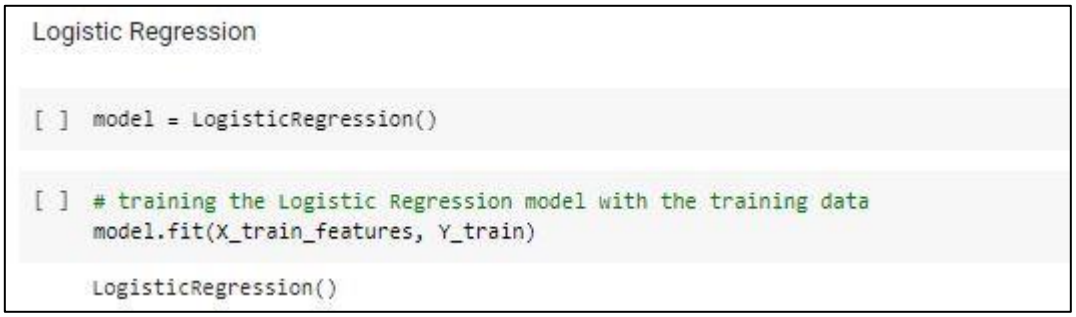

Figure 5.3: Screenshot of training logistic regression model.

Table (5.2) summarizes the findings obtained from the classification of the ham-spam dataset using logistic regression. The greatest result that was ever documented has an accuracy of 90.13 %, a precision of 98.29%, a sensitivity (Recall) of 90.10%, a specificity of 90.32%, and an F1-measure of 94.02%.

Table 5.2: Experimental results for logistic regression.

	Results				
Proposed model	Accuracy	Precision	Sensitivity	Specificity	FI- Score
Logistic regression	96.70 %	98.29 %	90.10 %	90.32 %	94.02%

The code that corresponds to the training accuracy result that we achieved is displayed in figure (5.4).

Evaluating the trained model

```
[ ]  # prediction on training data

     prediction_on_training_data = model.predict(X_train_features)
     accuracy_on_training_data = accuracy_score(Y_train, prediction_on_training_data)

[ ]  print('Accuracy on training data : ', accuracy_on_training_data)

     Accuracy on training data :  0.9670181736594121

[ ]  # prediction on test data

     prediction_on_test_data = model.predict(X_test_features)
     accuracy_on_test_data = accuracy_score(Y_test, prediction_on_test_data)

[ ]  print('Accuracy on test data : ', accuracy_on_test_data)

     Accuracy on test data :  0.9659192825112107
```

Figure 5.4: The screenshot illustrates the training accuracy code.

The confusion matrix for the testing data, which was generated by employing the proposed model, may be found displayed in figure (5.5).

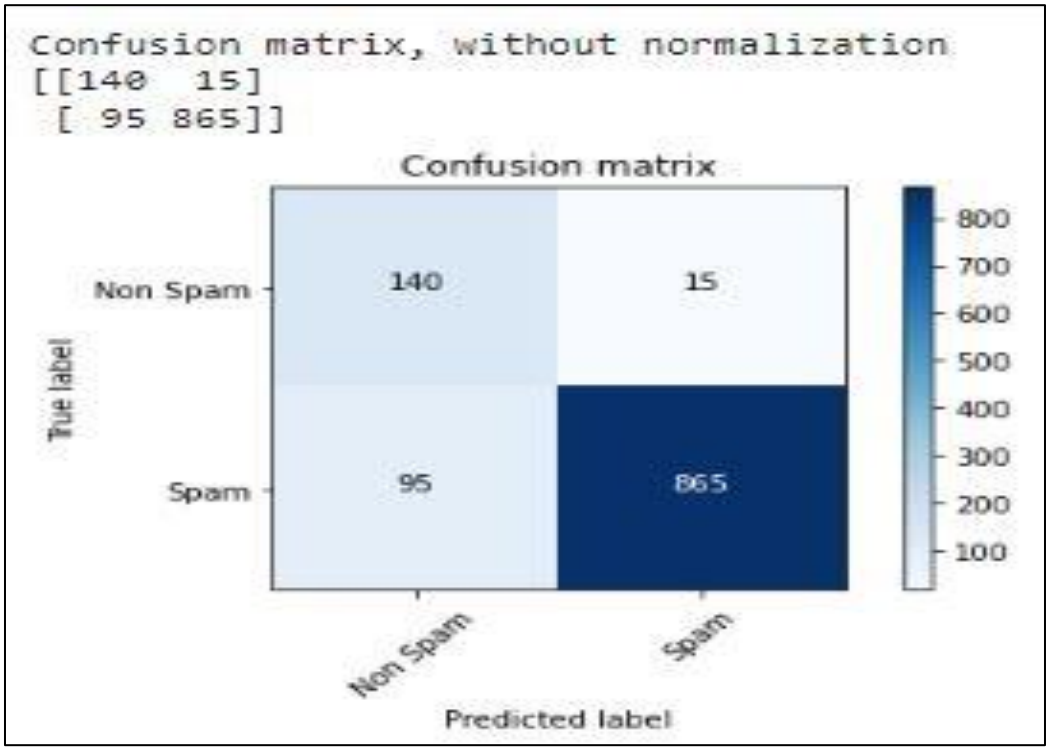

Figure 5.5: Experimental result of the confusion matrix.

The code that corresponds to the Metrics of the LR model's level of performance that we achieved is displayed in figure (5.6).

```
[ ] from sklearn.metrics import confusion_matrix
    from sklearn.metrics import confusion_matrix,f1_score, precision_score,recall_score

[ ] y_predict = clf.predict(x_test_features.toarray())

[ ] confusion_matrix(Y_test,y_predict)

    array([[140,  15],
           [ 95, 865]])

[ ] tn, fp, fn, tp = confusion_matrix(Y_test,y_predict).ravel()

[ ] print("Precision: {:.2f}%".format(100 * precision_score(Y_test, y_predict)))
    print("Recall: {:.2f}%".format(100 * recall_score(Y_test, y_predict)))

    Precision: 98.30%
    Recall: 90.10%

[ ] import matplotlib.pyplot as plt
    from sklearn.metrics import confusion_matrix
    import itertools

[ ] cnf_matrix = confusion_matrix(Y_test,y_predict)

[ ] def plot_confusion_matrix(cm, classes,
                              normalize=False,
                              title='Confusion matrix',
                              cmap=plt.cm.Blues):
        """
```

Figure 5.6: Screenshot of metrics of the LR model's level of performance.

The area under the curve (AUC) obtained from the experiment is shown in figure (5.7).

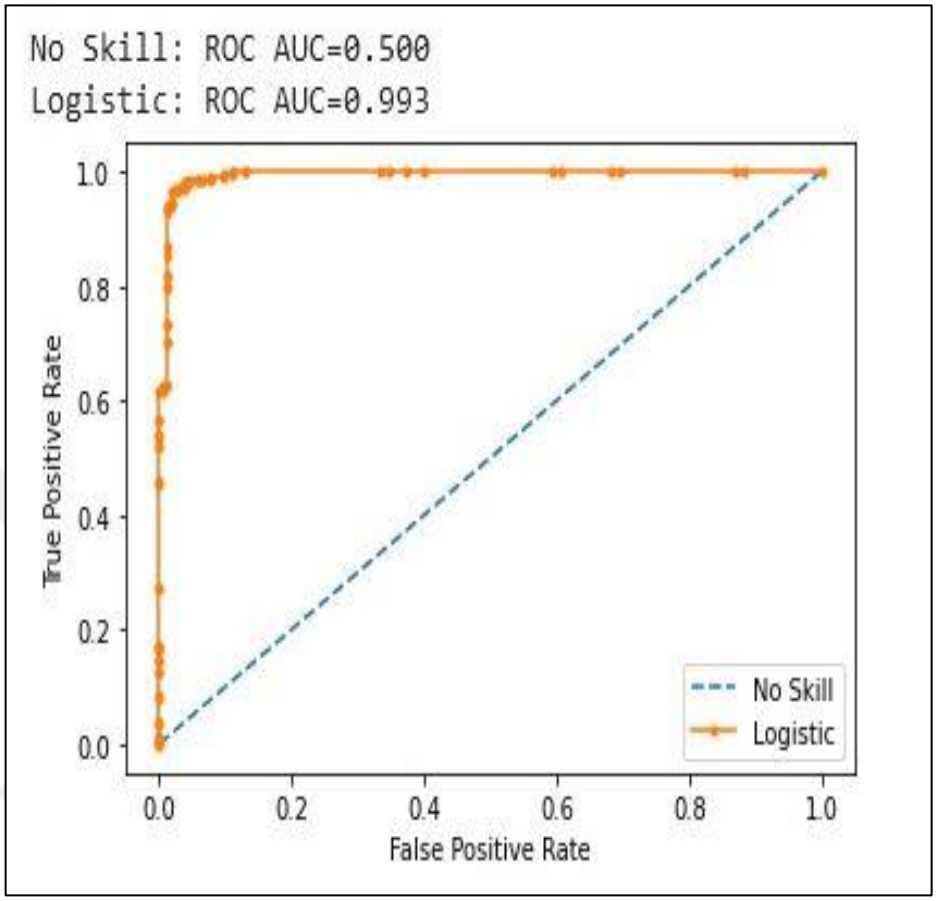

Figure 5.7: Logestic regression AUC plot.

5.5 RESULTS COMPARISON

We randomly selected older publications among those in which the authors' work focused on identifying spam by applying various machine learning techniques to a wide range of data sets. When we compared the findings to the proposed model, as shown in table (5.3), we discovered that the proposed model (LR) had achieved the highest level of accuracy compared to the other results. Figure (5.8) appears the comparison between the results through a histogram.

Table 5.3: Comparing our work to those of other authors.

Reference	Dataset	Classifier/Model/ Optimization	Results
Sharma, Vishnu Dutt, et al [82]	Email _small & Email_full	DT	Accuracy = 80.8% Precision = 88.2% Sensitivity = 94.4% Specificity = 80.6% FI-score = 91.2%
Mohammed et al. [83]	Email-1431	Naive Bayes	Accuracy = 85.96%
Agarwal & Kumar [84]	Ling-Spam	PSO- Naive Bayes	Accuracy = 90.50% Precision = 96.42% Sensitivity = 94.50% FI-score = 95.45%
Taloba & Ismail [85]	Enron	GA-DT	Accuracy = 90.5% Precision = 95.50% Sensitivity = 97.20% FI-score = 96.30%
Kumareson [86]	Ling-Spam	GA-SVM	Accuracy = 94.69% Precision = 98.52% Sensitivity = 20.12%
Faris et al. [87]	Spam Assasin	PSO-RF	Accuracy = 97.92%
Alghoul et al. [88]	-	ANN	Accuracy = 85.31%
Gomes et al. [89]	Enron	Hidden Markov model	Accuracy = 91.28%
Sharma et al. [90]	SpamBase	Random committee	Accuracy = 94.28%
Temitayo et al. [91]	Spam-Assasin	GA-SVM	Accuracy = 93.50%
Our proposed model	Ham-Spam dataset	Logistic regression	Accuracy = 96.70% Precision = 98.29% Sensitivity = 90.10% Specificity = 90.32% FI-score = 94.02%

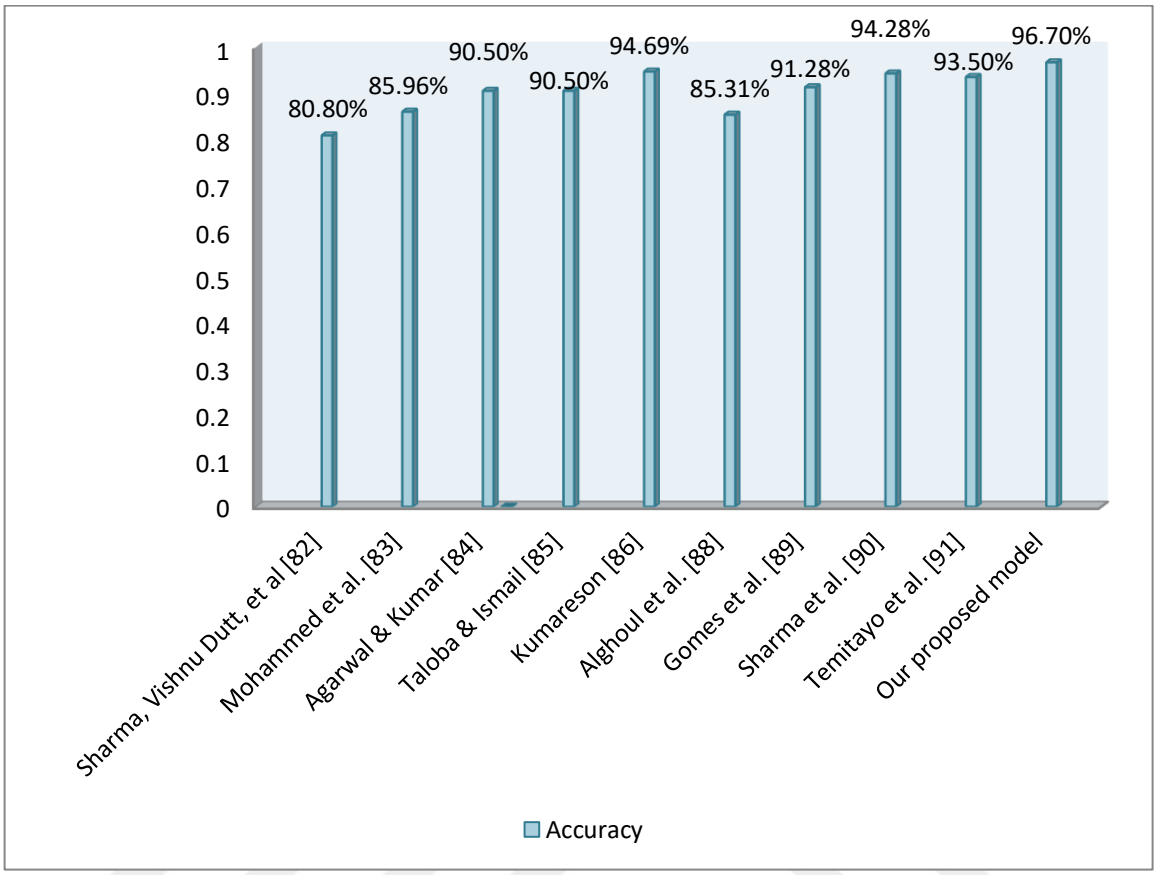

Figure 5.8: Comparison of results using the histogram.

6. DISCUSSION AND CONCLUSION

6.1 DISCUSSION

A significant number of researchers have used different approaches from the fields of machine learning and artificial intelligence in order to predict and categorize spam. These methods take data as their input, learn from it, and then make predictions on any new data that has the same dimension as the data they learn from in the future. The machine learning method known as logistic regression was utilized in the execution of this study. The sigmoid function is the one that is utilized for the activation function in the probabilistic statistical model known as logistic regression. This study uses the Ham-Spam dataset, which may be found in the open repository on the Kaggle site. The data consists of 5573 properties, with 4825 pertaining to the ham and 747 to the spam. The gaps in the data were filled in with a mean value derived from the aspects of the data that did not contain any missing values. For the classification job, the data are divided into three sets: the training set, which comprises 75% of the data; the testing set, which comprises 20% of the data; and the validation set, which constitutes 5% of the data.

Following the completion of the training of the logistic regression model with the help of the training set, the model that has been trained is next evaluated with the help of the test set. Under this body of work, we investigated the behavior of our model in scenarios in which the features of the data used for training and testing are scaled, as well as those in which the features of the data are not scaled.

6.2 CONCLUSION

Email is one of the most common methods of communication since it is readily available, the cost of sending messages is very low, and users can send and receive messages in a very short amount of time. A significant proportion of the currently available email spam filters are incapable of effectively preventing spam from accessing the user's inbox.

This is a result of the fact that spammers are continually developing more complex tactics that can easily circumvent spam filters.

Logistic regression is a model that does not depend on feature-level dependencies within a data set, does not have an extreme sensitivity to imbalanced data, and does not require

feature-level scaling of the data to function well. Therefore, individuals interested in classifying spam can tremendously benefit from adopting a logistic regression model to predict spam, which will go a long way toward controlling hackers on websites. This will go a long way toward controlling spam on websites as well. Because of its ease of use and reduced time commitment, logistic regression is an excellent choice for a probabilistic prediction model to use when working with data sets of medium size or larger. It is particularly useful for problems involving binary categorization.

In this study, we introduced a technique called Logistic regression model Induction for effectively and efficiently filtering email spam. Moreover, analyzed the performance of the LR algorithm on the ham-spam dataset by utilizing accuracy, precision, sensitivity, and specificity, as well as FI-measure, to select the algorithm that has the accuracy. We conclude our research that LR demonstrated potential for future success as an algorithm that can be used either at the mail server or the mail client side to minimize further the number of spam messages received in the inboxes of email users. To further evaluate the algorithm's effectiveness in preventing spam, we plan to use it in the future on more email spam datasets that are accessible to the general public**.**

REFERENCES

[1] Delany, S. J., Buckley, M., & Greene, D. (2012). SMS spam filtering: Methods and data. Expert Systems with Applications, 39(10), 9899-9908.

[2] Diale, Melvin, Turgay Celik, and Christiaan Van Der Walt. "Unsupervised feature learning for spam email filtering." Computers & Electrical Engineering 74 (2019): 89-104.

[3] Kigerl, Alex. "Cyber Crime Nation Typologies: K-Means Clustering of Countries Based on Cyber Crime Rates." International Journal of Cyber Criminology 10.2(2016).

[4] Ala'M, Al-Zoubi, et al. "Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts." Knowledge-Based Systems 153 (2018): 91-104.

[5] W. Feng, J. Sun, L. Zhang, C. Cao, and Q. Yang, ``A support vector machine based Naive Bayes algorithm for spam _ltering," in Proc. IEEE 35th Int. Perform. Comput. Commun. Conf. (IPCCC), Dec. 2016, pp. 1_8, doi:10.1109/pccc.2016.7820655.

[6] Dada, Emmanuel Gbenga, et al. "Machine learning for email spam filtering: review, approaches and open research problems." Heliyon 5.6 (2019): e01802.

[7] Sun, Guanglu, et al. "Active learning method for chinese spam filtering." International Journal of Performability Engineering 13.4 (2017): 511.

[8] Guzella, Thiago S., and Walmir M. Caminhas. "A review of machine learningapproaches to spam filtering." Expert Systems with Applications 36.7 (2009):10206-10222.

[9] Tida, Vijay Srinivas, and Sonya Hsu. "Universal Spam Detection using Transfer Learning of BERT Model." arXiv preprint arXiv:2202.03480 (2022).

[10] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Journal of machine Learning research 3.Jan (2003): 993-1022.

[11] Surwade, Ajay U. "Phishing email is an increasing menace." International Journal of Information Technology 12.2 (2020): 611-617.

[12] Almomani, Ammar, et al. "A survey of phishing email filtering techniques." IEEE communications surveys & tutorials 15.4 (2013): 2070-2090.

[13] Kaur, Gurwinder, et al. "A Survey on Various Classification Techniques in Email Spamming." International Journal of Technology and Computing (IJTC) 5.3(2016):589-593.

[14] Zeidanloo, Hossein Rouhani, et al. "A taxonomy of botnet detection techniques." 2010 3rd International Conference on Computer Science and Information Technology. Vol. 2. IEEE, 2010.

[15] Jacob, Grégoire, Hervé Debar, and Eric Filiol. "Behavioral detection of malware: from a survey towards an established taxonomy." Journal in computer Virology 4.3 (2008):251-266.

[16] Rafique, M. Zubair, et al. "Evolutionary algorithms for classification of malware families through different network behaviors." Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. 2014.

[17] Nappa, Antonio, M. Zubair Rafique, and Juan Caballero. "The MALICIA dataset: identification and analysis of drive-by download operations." International Journal of Information Security 14.1 (2015): 15-33.

[18] Bulazel, Alexei, and Bülent Yener. "A survey on automated dynamic malware analysis evasion and counter-evasion: Pc, mobile, and web." Proceedings of the 1st Reversing and Offensive-oriented Trends Symposium. 2017.

[19] El Merabet, Hoda, and Abderrahmane Hajraoui. "A survey of malware detection techniques based on machine learning." International Journal of Advanced Computer Science and Applications 10.1 (2019).

[20] Petersen, Line Nybro. "The ageing body in monty Python live (mostly)." European journal of cultural studies 21.3 (2018): 382-394.

[21] J. Dunagan, D. R. Simon, H. J. Wang, and U. C. Berkeley, "Characterizing Botnets from Email Spam Records," LEET, vol. 8, pp. 1--9, 2018.

[22] Lee, Daekyung, Mi Jin Lee, and Beom Jun Kim. "Deviation-based spam-filtering method via stochastic approach." EPL (Europhysics Letters) 121.6 (2018): 68004.

[23] Jain, Ankit Kumar, and Brij B. Gupta. "Towards detection of phishing websites on client-side using machine learning based approach." Telecommunication Systems 68.4 (2018): 687-700.

[24] Nataraj, Lakshmanan, and B. S. Manjunath. "Spam: Signal processing to analyze malware [applications corner]." IEEE Signal Processing Magazine 33.2 (2016): 105-117.

[25] Ruff, Lukas, et al. "A unifying review of deep and shallow anomaly detection." Proceedings of the IEEE 109.5 (2021): 756-795.

[26] Santos, Igor, et al. "Spam filtering through anomaly detection." International Conference on E-Business and Telecommunications. Springer, Berlin, Heidelberg, 2011.

[27] Satapathy, S., et al. "Intelligent Data Engineering and Analytics." J Advances in Intelligent Systems and Computing 2 (2021): 1177.

[28] Cidon, Asaf, et al. "High precision detection of business email compromise." 28th USENIX Security Symposium (USENIX Security 19). 2019.

[29] Karim, Asif, et al. "A comprehensive survey for intelligent spam email detection." IEEE Access 7 (2019): 168261-168295.

[30] Goodall, John R., Wayne G. Lutters, and Anita Komlodi. "I know my network: collaboration and expertise in intrusion detection." Proceedings of the 2004 ACMconference on Computer supported cooperative work. 2004.

[31] Bahnsen, Alejandro Correa, et al. "Classifying phishing URLs using recurrent neural networks." 2017 APWG symposium on electronic crime research (eCrime). IEEE, 2017.

[32] Laorden, Carlos, et al. "Study on the effectiveness of anomaly detection for spam filtering." Information Sciences 277 (2014): 421-444.

[33] Stolfo, Salvatore J., Ke Wang, and Wei-Jen Li. "Fileprint analysis for malware detection." ACM CCS WORM (2005).

[34] Fereidooni, Hossein, et al. "Breaking fitness records without moving: Reverse engineering and spoofing fitbit." International Symposium on Research in Attacks,Intrusions, and Defenses. Springer, Cham, 2017.

[35] Alwaghid, Alhanoof Faiz, and Nurul I. Sarkar. "Exploring malware behavior of webpages using machine learning technique: An empirical study." Electronics 9.6 (2020): 1033.

[36] Pandey, Alok, and Jatinderkumar R. Saini. "Attacks & defense mechanisms for TCP/IP based protocols." International Journal of Engineering Innovations andResearch 3.1 (2014): 17.

[37] Li, Xingguo, Junfeng Wang, and Xiaosong Zhang. "Botnet detection technology based on DNS." Future Internet 9.4 (2017): 55.

[38] Al-Garadi, Mohammed Ali, et al. "A survey of machine and deep learning methods for internet of things (IoT) security." IEEE Communications Surveys &Tutorials 22.3(2020): 1646-1685

[39] Christina, V., S. Karpagavalli, and G. Suganya. "Email spam filtering using supervised machine learning techniques." International Journal on Computer Science and Engineering (IJCSE) 2.09 (2010): 3126-3129.

[40] Christina, V., S. Karpagavalli, and G. Suganya. "A study on email spam filteringtechniques." International Journal of Computer Applications 12.1 (2010).

[41] Saad, Omar, Ashraf Darwish, and Ramadan Faraj. "A survey of machine learning techniques for Spam filtering." International Journal of Computer Science and Network Security (IJCSNS) 12.2 (2012): 66.

[42] Hussain, Naveed, et al. "Spam review detection using the linguistic and spammer behavioral methods." IEEE Access 8 (2020): 53801-53816.

[43] Lim, Ee-Peng, et al. "Detecting product review spammers using rating behaviors." Proceedings of the 19th ACM international conference on Information and knowledge management. (2010).

[44] Liu, Yuanchao, Bo Pang, and Xiaolong Wang. "Opinion spam detection by incorporating multimodal embedded representation into a probabilistic reviewgraph." Neurocomputing 366 (2019): 276-283.

[45] Pujiarto, Bambang, et al. "A data mining practical approach to inventory management and logistics optimization." International Journal of Informatics and Information Systems 4.2 (2021): 112-122.

[46] Dhanaraj, S., and V. Karthikeyani. "A study on email image spam filtering techniques." 2013 international conference on pattern recognition, informatics and mobile engineering. IEEE, 2013.

[47] Bhowmick, Alexy, and Shyamanta M. Hazarika. "Email spam filtering: a review of techniques and trends." Advances in electronics, communication andcomputing (2018): 583-590.

[48] Li, Xingguo, Junfeng Wang, and Xiaosong Zhang. "Botnet detection technology based on DNS." Future Internet 9.4 (2017): 55.

[49] Al-Garadi, Mohammed Ali, et al. "A survey of machine and deep learning methods for internet of things (IoT) security." IEEE Communications Surveys & Tutorials 22.3(2020): 1646-1685.

[50] Sanz, Enrique Puertas, José María Gómez Hidalgo, and José Carlos Cortizo Pérez. "Email spam filtering." Advances in computers 74 (2008): 45-114.

[51] Christina, V., S. Karpagavalli, and G. Suganya. "Email spam filtering using supervised machine learning techniques." International Journal on Computer Science and Engineering (IJCSE) 2.09 (2010): 3126-3129.

[52] Saad, Omar, Ashraf Darwish, and Ramadan Faraj. "A survey of machine learning techniques for Spam filtering." International Journal of Computer Science and Network Security (IJCSNS) 12.2 (2012): 66.

[53] Lim, Ee-Peng, et al. "Detecting product review spammers using rating behaviors." Proceedings of the 19th ACM international conference on Information and knowledge management. 2010.

[54] Garera, Sujata, et al. "A framework for detection and measurement of phishing attacks." Proceedings of the 2007 ACM workshop on Recurring malcode. 2007.

[55] Zareapoor, Masoumeh, and K. R. Seeja. "Feature extraction or feature selection for text classification: A case study on phishing email detection." International Journal of Information Engineering and Electronic Business 7.2 (2015): 60.

[56] Fette, Ian, Norman Sadeh, and Anthony Tomasic. "Learning to detect phishing emails." Proceedings of the 16th international conference on World Wide Web. 2007.

[57] Toolan, Fergus, and Joe Carthy. "Feature selection for spam and phishing detection." 2010 eCrime Researchers Summit. IEEE, 2010.

[58] Gallo, Luigi, Alessio Botta, and Giorgio Ventre. "Identifying threats in a largecompany's inbox." Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks. 2019.

[59] Cidon, Asaf, et al. "High precision detection of business email compromise." 28th USENIX Security Symposium (USENIX Security 19). 2019.

[60] Karim, Asif, et al. "A comprehensive survey for intelligent spam email detection." IEEE Access 7 (2019): 168261-168295.

[61] Nandhini, S., and Jeen Marseline KS. "Performance evaluation of machine learning algorithms for email spam detection." 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE). IEEE, 2020.

[62] GuangJun, Luo, et al. "Spam detection approach for secure mobile message communication using machine learning algorithms." Security and CommunicationNetworks 2020 (2020).

[63] Siddique, Zeeshan Bin, et al. "Machine learning-based detection of spam emails." Scientific Programming 2021 (2021).

[64] Poomka, Pumrapee, et al. "SMS spam detection based on long short-term memory and gated recurrent unit." International Journal of Future Computer andCommunication 8.1 (2019): 11-15.

[65] Rajamohana, S. P., K. Umamaheswari, and B. Abirami. "Adaptive binary flower pollination algorithm for feature selection in review spam detection." 2017 (IGEHT). IEEE, 2017.

[66] Khan, Muhammad Saad Shahbaz, et al. "Variable Generalization Evaluation of Supervised Learning Models for Detection of Spam Messages." 2021 International Conference on Innovative Computing (ICIC). IEEE, 2021.

[67] Nandhini, S., and Jeen Marseline KS. "Performance evaluation of machine learning algorithms for email spam detection." 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE). IEEE, 2020.

[68] Mai, Xiaoyi, Zhenyu Liao, and Romain Couillet. "A large scale analysis of logistic regression: Asymptotic performance and new insights." ICASSP 2019-2019 IEEEInternational Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019.

[69] Lynam, Anita L., et al. "Logistic regression has similar performance to optimised machine learning algorithms in a clinical setting: application to the discrimination between type 1 and type 2 diabetes in young adults." Diagnostic and prognostic research 4.1 (2020): 1-10.

[70] Coxe, Stefany, Stephen G. West, and Leona S. Aiken. "Generalized linear models." The Oxford handbook of quantitative methods 2 (2013): 26-51.

[71] Zou, X., Hu, Y., Tian, Z., & Shen, K. (2019, October). Logistic regression model optimization and case analysis. In 2019 IEEE 7th international conference on computer science and network technology (ICCSNT) (pp. 135-139). IEEE.

[72] Levy, J. J., & O'Malley, A. J. (2020). Don't dismiss logistic regression: the case for sensible extraction of interactions in the era of machine learning. BMC medical research methodology, 20(1), 1-15.

[73] Jhajharia, S., Varshney, H. K., Verma, S., & Kumar, R. (2016, September). A neural network based breast cancer prognosis model with PCA processed features. In 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1896-1901). IEEE.

[74] Khan, M. R. H., Afroz, U. S., Masum, A. K. M., Abujar, S., & Hossain, S. A. (2020, July). Sentiment analysis from bengali depression dataset using machine learning. In 2020 11th international conference on computing, communication and networking technologies (ICCCNT) (pp. 1-5). IEEE.

[75] Mostafa, G., Ahmed, I., & Junayed, M. S. (2021). Investigation of different machine learning algorithms to determine human sentiment using Twitter data. International Journal of Information Technology and Computer Science, 13(2), 38-48.

[76] Patel, R., & Passi, K. (2020). Sentiment analysis on twitter data of world cup soccer tournament using machine learning. IoT, 1(2), 14.

[77] Rapacz, S., Chołda, P., & Natkaniec, M. (2021). A method for fast selection of machine-learning classifiers for spam filtering. Electronics, 10(17), 2083.

[78] Al Moubayed, N., Breckon, T., Matthews, P., & McGough, A. S. (2016). Sms spam filtering using probabilistic topic modelling and stacked denoising autoencoder. In Artificial Neural Networks and Machine Learning–ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II 25 (pp. 423-430). Springer International Publishing.

[79] Saez, A., Acha, B., & Serrano, C. (2014). Pattern analysis in dermoscopic images. Computer vision techniques for the diagnosis of skin Cancer, 23-48.

[80] Deng, L., & Yu, D. (2014). Deep learning: methods and applications. Foundations and trends® in signal processing, 7(3–4), 197-387.

[81] Meyer, A., Zverinski, D., Pfahringer, B., Kempfert, J., Kuehne, T., Sündermann, S. H., ... & Eickhoff, C. (2018). Machine learning for real-time prediction of complications in critical care: a retrospective study. The Lancet Respiratory Medicine, 6(12), 905-914.

[82] Sharma, V. D., Yadav, S. K., Yadav, S. K., Singh, K. N., & Sharma, S. (2021). An effective approach to protect social media account from spam mail–a machine learning approach. Materials Today: Proceedings, 2(3), 1491.

[83] Mohammed, S., Mohammed, O., Fiaidhi, J., Fong, S., & Kim, T. H. (2013). Classifying unsolicited bulk email (UBE) using python machine learning techniques. International Journal of Hybrid Information Technology, 6(1), 43-56.

[84] Agarwal, K., & Kumar, T. (2018, June). Email spam detection using integrated approach of Naïve Bayes and particle swarm optimization. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 685-690). IEEE.

[85] Taloba, A. I., & Ismail, S. S. (2019, December). An intelligent hybrid technique of decision tree and genetic algorithm for e-mail spam detection. In 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS) (pp. 99-104). IEEE.

[86] Kumareson, T. (2016). Certain Investigations On Optimization Techniques to Enhance E-Mail Spam Classification. Anna University. Accessed: Feb, 26, 2020.

[87] Faris, Hossam, Ibrahim Aljarah, and Bashar Al-Shboul. "A hybrid approach based on particle swarm optimization and random forests for email spam filtering." International conference on computational collective intelligence. Springer, Cham, 2016.

[88] Alghoul, Ahmed, et al. "Email classification using artificial neural network." IJARW, (2018).

[89] Gomes, Sebastian Romy, et al. "A comparative approach to email classification using Naive Bayes classifier and hidden Markov model." 2017 4th internationalconference on advances in Electrical Engineering (ICAEE). IEEE, 2017.

[90] Sharma, Sumant, and Amit Arora. "Adaptive approach for spam detection." International Journal of Computer Science Issues (IJCSI) 10.4 (2013).

[91] Temitayo, Fagbola, Olabiyisi Stephen, and Adigun Abimbola. "Hybrid GA-SVM for efficient feature selection in email classification." Computer Engineering and Intelligent Systems 3.3 (2012): 17-28.