REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

PHISHING, SPAM AND MALWARE DETECTION IN EMAILS USING DEEP LEARNING AND ANOMALY DETECTION FILTERS

Mays Akram Fadhil FADHIL

Master's Thesis

Supervisor

Prof. Dr. Osman Nuri UÇAN

Istanbul, 2022

PHISHING, SPAM AND MALWARE DETECTION IN EMAILS USING DEEP LEARNING AND ANOMALY DETECTION FILTERS

Mays Akram Fadhıl FADHIL

Electrical and Computer Engineering

Master's Thesis

ALTINBAŞ UNIVERSITY

2022

The thesis titled PHISHING, SPAM AND MALWARE DETECTION IN EMAILS USING DEEP LEARNING AND ANOMALY DETECTION FILTERS prepared by MAYS AKRAM FADHIL FADHI and submitted on 20/12 /2022 has been **accepted unanimously** for the degree of Master of Science in Electrical and Computer Engineering.

Prof. Dr. Osman Nuri UÇAN

Supervisor

Thesis Defense Committee Members:

Prof. Dr. Osman Nuri UÇAN	Department of Electrical-Electronics Engineering, Altınbaş University	_____
Asst.Prof.Dr.Sefer Kurnaz	Department of Computer Engineering, Altınbaş University	_____
Assoc.prof. Dr. Adil Deniz DURU	Department of Sports and Health Sciences, 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 Institute of Graduate Studies: ___/___/___

I hereby declare that all information and 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.

MAYS FADHIL

Signature

DEDICATION

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

PREFACE

First of all, I would like to extend my gratitude and thanks to my supervisor, Prof. Dr. Osman Nuri UÇAN, for supporting me in writing this scientific thesis. This helped me understand the main reason behind writing the thesis. And I was able to know the technical and scientific need for this research work.

ABSTRACT

PHISHING, SPAM AND MALWARE DETECTION IN EMAILS USING DEEP LEARNING AND ANOMALY DETECTION FILTERS

Mays Akram Fadhil FADHIL

M.Sc., Electrical and Computer Engineering, Altınbaş University,
Supervisor: Prof. Dr. Osman Nuri UÇAN

Date: December/2022

Pages: 71

Spam is an unsolicited electronic message, sent massively to a large number of recipients, for advertising or malicious purposes. The term spam is also used to designate the same type of message transmitted by other means of electronic communication such as instant messaging, blogs, forums, and more recently, mobile telephone networks, via SMS or MMS. Even if the means of communication is different, the techniques of sending and detection remain relatively similar In this thesis we propose the use of two different filter methods for spam and phishing detection and then comparing the accuracies of these detection methods with other methods such as decision tree and ANN, these filters work on learning from data also called (machine learning,

Keywords: ANN, SVM, ML, DL, Spam.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

AI	:	Artificial Intelligence
FSK	:	Frequency Shift Keying
IoT	:	Internet of Things
PAN	:	Personal Area Network
OFDM	:	Orthogonal Frequency Division Multiplexing
TDMA	:	Time Division Multiple Access

1. INTRODUCTION

1.1 INTRODUCTION

Spam is a big problem for Internet users. Recent increases in the spam rate have caused great concern among the Internet community. Many solutions had been suggested to solve the problem. In this chapter, we first present the beginnings of spam, its objectives, its contents, its impacts and the different techniques used to detect this type of email.

1.2 PROBLEM STATEMENT

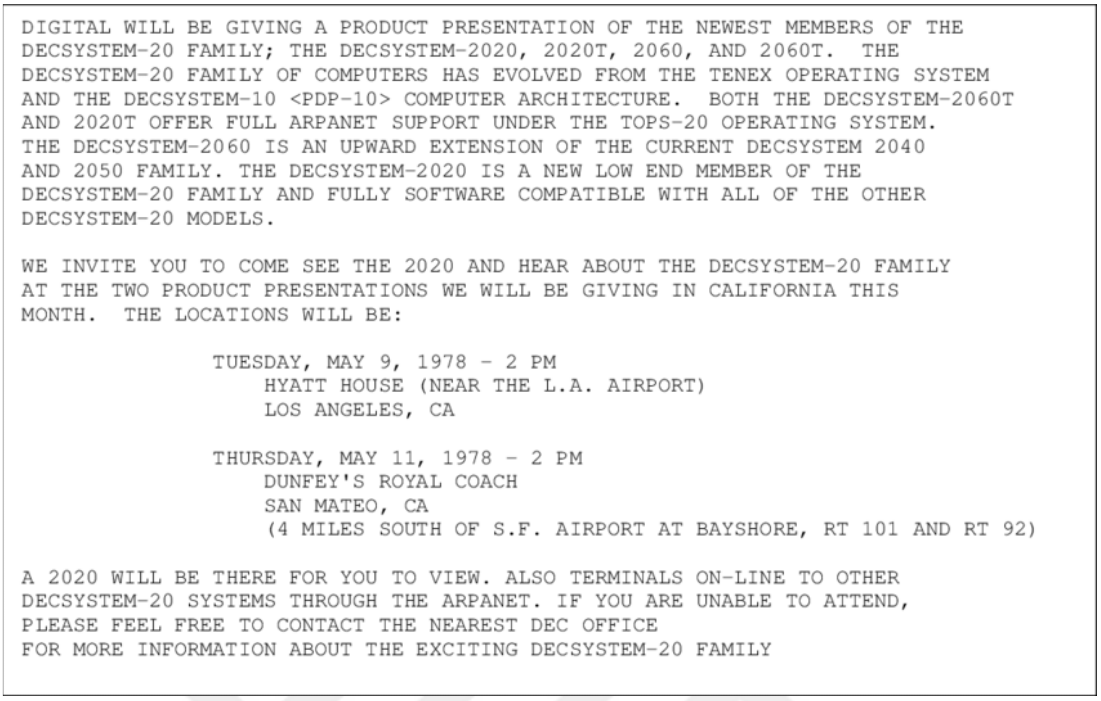
1.2.1 Origin of the Word Spam

In 1937 Hormel Foods organized a competition to find a new name for their spiced ham, this name must be as characteristic as the taste of the product "Spiced Ham" and who offers "Spam" for this product, was therefore the brand chosen. This pre-cooked meat box often synonymous with bad food was widely used by the United States armed forces quartermaster for the food of soldiers during the World War 2 and will be introduced in various regions of the world on this occasion.

1.2.2 Definition of Spam

Spam is an unsolicited electronic message, sent massively to a large number of recipients, for advertising or malicious purposes. The term spam is also used to designate the same type of message transmitted by other means of electronic communication such as instant messaging, blogs, forums, and more recently, mobile telephone networks, via SMS or MMS. Even if the means of communication is different, the techniques of sending and detection remain relatively similar. The first spam (Figure 1.1) dates from May 3, 1978. That day, on the ARPANET network, Gary Thuerk, a sales representative from the computer company DEC3, invited 393 people by e-mail to discover his new machine, the 2020.

The message looked like this:



DIGITAL WILL BE GIVING A PRODUCT PRESENTATION OF THE NEWEST MEMBERS OF THE
DECSYSTEM-20 FAMILY; THE DECSYSTEM-2020, 2020T, 2060, AND 2060T. THE
DECSYSTEM-20 FAMILY OF COMPUTERS HAS EVOLVED FROM THE TENEX OPERATING SYSTEM
AND THE DECSYSTEM-10 <PDP-10> COMPUTER ARCHITECTURE. BOTH THE DECSYSTEM-2060T
AND 2020T OFFER FULL ARPANET SUPPORT UNDER THE TOPS-20 OPERATING SYSTEM.
THE DECSYSTEM-2060 IS AN UPWARD EXTENSION OF THE CURRENT DECSYSTEM 2040
AND 2050 FAMILY. THE DECSYSTEM-2020 IS A NEW LOW END MEMBER OF THE
DECSYSTEM-20 FAMILY AND FULLY SOFTWARE COMPATIBLE WITH ALL OF THE OTHER
DECSYSTEM-20 MODELS.

WE INVITE YOU TO COME SEE THE 2020 AND HEAR ABOUT THE DECSYSTEM-20 FAMILY
AT THE TWO PRODUCT PRESENTATIONS WE WILL BE GIVING IN CALIFORNIA THIS
MONTH. THE LOCATIONS WILL BE:

TUESDAY, MAY 9, 1978 - 2 PM
HYATT HOUSE (NEAR THE L.A. AIRPORT)
LOS ANGELES, CA

THURSDAY, MAY 11, 1978 - 2 PM
DUNFEY'S ROYAL COACH
SAN MATEO, CA
(4 MILES SOUTH OF S.F. AIRPORT AT BAYSHORE, RT 101 AND RT 92)

A 2020 WILL BE THERE FOR YOU TO VIEW. ALSO TERMINALS ON-LINE TO OTHER
DECSYSTEM-20 SYSTEMS THROUGH THE ARPANET. IF YOU ARE UNABLE TO ATTEND,
PLEASE FEEL FREE TO CONTACT THE NEAREST DEC OFFICE
FOR MORE INFORMATION ABOUT THE EXCITING DECSYSTEM-20 FAMILY

Figure 1.1: The first spam

This unwanted message was unfortunately only the first of a long series. Spam was born. [4]

1.3 SPAM GOALS AND STATISTICS

Initially, spam was primarily aimed at advertising purposes. Today, it has grown considerably, diversified and complexified, to more and more often achieve malevolent objectives. Indeed, Spam has not only grown in terms of volume, but also in terms of content (see figure 1.2). Today, the objectives of spam are very varied here is a non-exhaustive list:

a. Phishing (or phishing): The objective is to succeed in impersonating an organization known to the user, in order to steal confidential information. For example, we receive an email "apparently" from our bank, or from another site where we have personal information. in this email, you are asked to click on a link (for various reasons: updating, etc.), after clicking on this link, a web page is displayed... on which you are asked to enter your bank details or any other personal information. Some of the top most counterfeited sites for phishing attacks include eBay, Paypal, and Bank of America.

b. Advertising: The objective is to extol the merits of any product. These include, for example, pharmaceutical products, luxury products, various and varied software,

gambling. They may also support political, cultural or religious ideas and/or organizations.

c. Scams: This is an attack based on the naivety of the recipients with the aim of extracting money from them. The most common example is the Nigerian scam: a dignitary from an African country asks you to act as an intermediary for a large financial transaction, promising you a good percentage of the sum. To initiate the transaction, you need to give money. [6]

d. Prank: The objective is to circulate information that seems very sensitive, often with an urgency character: false virus alert, false alert of potential contamination, chain of solidarity…. For example: “a very dangerous new virus is spreading, information must be circulated”; “underclothes are infected with a dangerous bacterium”.

e. Malware: Is software designed to infiltrate or damage a computer system. It is commonly taken to contain computer viruses, worms, Trojan horses, spyware and adware. This type of software is often sent as a non-suspicious email attachment. When the user opens the file, the malware installs. The interrelationship between spam and malware has evolved Spam malware spreads e-mails, malware is used to infect a host so that the host can be controlled remotely and used for sending over Spam. These infected hosts are referred to as 'zombie computers'. Many people believe that most spam is sent by botnets, which are a network of bot PCs. [7]

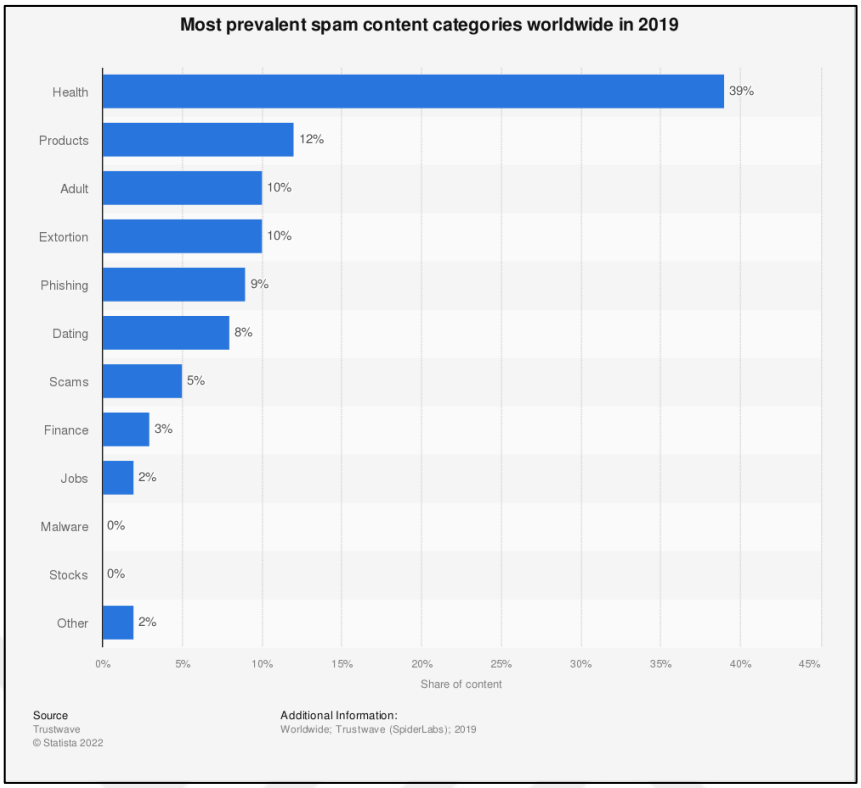


Figure 1.2: Breakdown of spam by content [8]

We have some statistics on the overall spam rate between the years 2012 and 2017 presented in Figure 1.4. In the last period it was found that spam accounted for 55% of all e-mail messages, as in the previous year.

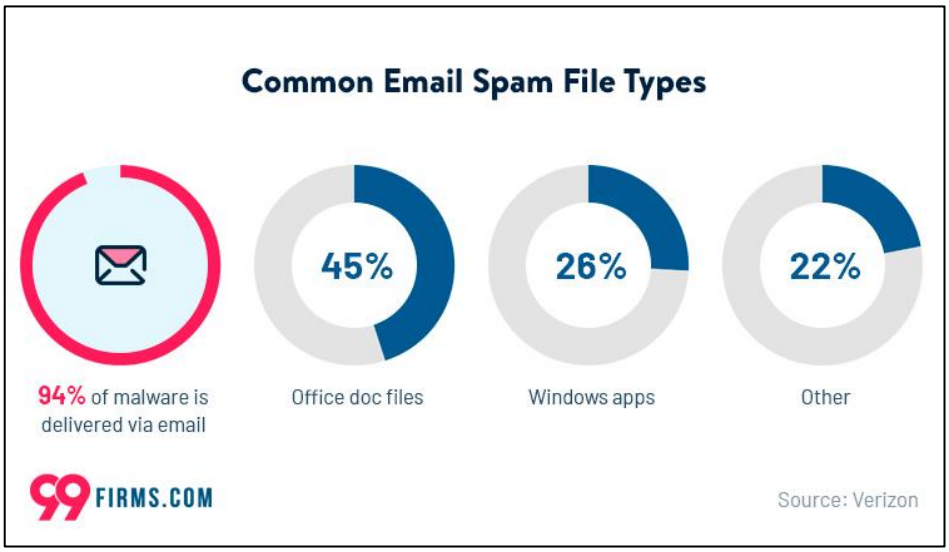


Figure 1.3: spam development in terms of volume [8]

1.4 IMPACTS OF SPAM ON USERS AND PROVIDERS

In this section, we present the effects of spam, at the level of users, companies and ISPs. [9]

1.4.1 Waste of Time

a. Abnormal congestion of mailboxes.

b. Delete unwanted emails.

c. Configuration and maintenance of filters.

d. Consultation of rejected emails to detect the good ones because of the risk of missing important emails that are poorly catalogued by anti-spam detection tools.

1.4.2 Loss of Bandwidth and Disk Space

a. Especially for modem users.

b. Virus and spam attachments can be large.

1.4.3 Significant Financial Losses at Corporate and ISP Levels

a. an increase in operational management and support costs related to anti-spam management.

b. loss of employee productivity.

According to one study, spam costs companies approximately $712 per employee per year. To this figure, we must add $113 to $183 per employee per year for the management of quarantined emails.

1.5 SPAM FILTERING TECHNIQUES

Several anti-spam techniques are possible and can be combined: statistical analysis (Bayesian filter), filtering by keywords, white lists, black lists. These countermeasures must constantly adapt as new types of spam manage to circumvent them. Two spam detection solutions are possible: detection at the level of the ISP mail server and detection at the level of the end user.

These tools can be divided into two groups: envelope filtering, and content filtering.

1.5.1 Envelope Filtering

This type of filtering applies only to the header of the message, which often contains enough information to be able to distinguish spam. This technique applied at the ISP server level has the advantage of being able to block emails even before their body is sent, which greatly reduces traffic on the SMTP gateway.

In this category we find the following techniques:

a. Filtering by blacklists:

These lists consist of pre-declaring a list of “bad senders” (email addresses, domain names, countries, IP addresses), from which the recipient refuses to receive emails.

These lists can be:

i. created by the administrator or the user.

ii. downloaded via the web (this requires very regular updating for optimized filtering)

iii. Consulted in real time on the web (RBL, Real Time Blackhole List).

To circumvent these lists, spammers frequently change their sender addresses (email or IP). [10]

b. Filtering by whitelists:

These lists consist of pre-declaring a list of (email addresses, domain names, IP addresses) that the recipient agrees to receive emails from. By default very few hosts are considered safe because their addresses could be spoofed by spammers. Just like the blacklist, the whitelist also needs continuous upgrading and refreshing. [11]

c. Filtering by gray list:

The gray list is a mix between the white list and the black list. What happens is that each time a given mailbox receives an email from an unknown contact, this email is suspended with an automatic response message containing a link to validate the sending. This is to detect bots, spammers won't realize that they need to issue validation in order for the message to be accepted. In the case of a real expected email and the sender is not listed in one or the other of the black and white lists, then it will be positioned in the gray list. If the sender satisfies the confirmation request (often a web link to click), then they will get the move to whitelist and its messages will be forwarded to you. It is actually the dynamic opening of the whitelist.

For example: the following figure shows a response of this technique.

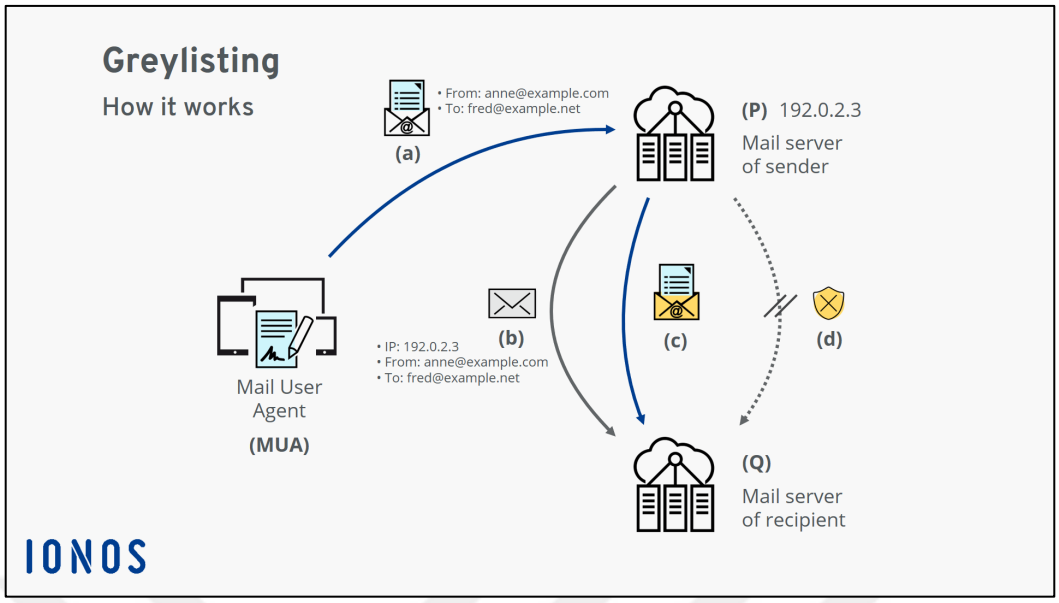


Figure 1.4: Example on a greylist technique response [10]

d. Filtering by domain verification:

Recipients are configured to only accept messages from specific domains. Emails with domains not listed will not be received. This way a lot of spam is blocked. [12]

1.5.2 Content Filtering

This type of filtering is done at the user level where their content is analyzed to detect spam that has managed to pass through the envelope filter.

In this category we find the following techniques:

a. Filtering by keywords:

The administrator must indicate the list of keywords to detect in order to determine that an email is spam. For example, all emails that contain the words: Viagra, money, money, drugs will be detected as Spam.

This filter is based on the keywords included in the emails. The analysis is very fast, but not effective. Because it requires manual monitoring and Spammers vary the keywords to avoid this filter. For example, we find MONEY or even m*o*n*e*y. [13]

b. Filtering by characters:

This is to block emails that contain certain characters or fonts, or certain languages used in these emails.

c. Image filtering:

This involves analyzing the images obtained in the messages at the level of the properties of the image file (format, file size, image size) and of the content of the image (colors, pixel test, etc.).

d. URL filtering:

This consists of checking the hypertext links included in the messages against a database of pre-recorded “bad URLs”, or via real-time consultation of the blacklists available on the web. Attempts to mask the hyperlink are sometimes used by spammers to prevent analysis by URL filtering.

1.6 CONTRIBUTION

In this thesis we propose the use of two different filter methods for spam and phishing detection and then comparing the accuracies of these detection methods with other methods such as decision tree and ANN, these filters work on learning from data also called (machine learning, these filters we mention namely:

1.6.1 Bayesian Filters

The best-known machine learning approach in spam filtering is the Naive Bayes classifier, Naive Bayes classifier is a probabilistic classifier. In short, it calculates and uses the probability of certain words/expressions appearing in the most known examples (posts) in order to rank new examples (posts). Naïve Bayes has been shown to be very successful at categorizing text documents. Bayesian filters (statistical method) Filters worked by analyzing the words of the message inside an email to calculate the probability that the message is spam or not. Calculation based on words that determine the message is spam and words that determine the message is not spam.

1.6.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) have had success in classifying text documents. SVM has given rise to significant research in applying them to spam filtering. SVM are kernel methods whose central idea is to integrate data representing text documents into a vector space. SVM attempt to construct a linear separation between two classes in this vector space. A support vector machine is a maximum-margin binary linear classifier. It can be interpreted as finding a hyperplane in a linearly separable feature space that separates the two classes with a maximum margin. The instances closest to the hyperplane are known as the "support vectors" because they support the hyperplane on both sides of the margin. SVM has been

reported to perform significantly on the text categorization problem with many relevant features. SVM has also been applied to spam filtering. [7]

1.7 CONCLUSION

In this chapter we present the definitions of spam, its objectives and impacts as well as the different approaches to defeating spam by little dividing it into two categories: The first contains email header-based solutions such as blacklists, white and gray. The second group of solutions contains those that are based on the textual content of the message such as filtering based on machine learning.

2. LITERATURE REVIEW

2.1 CHAPTER INTRODUCTION

The importance of ensuring that one's online presence is both trustworthy and safe has progressively grown over the course of the past several years. Internet, mobile computing, and electronic media are the building blocks upon which the majority of the services that we rely on on a daily basis are constructed. One method of communication is email, and as the internet has become more widespread, the number of people who send and receive emails has also increased. It is one of the simplest forms of criminal conduct that may be carried out via email, and that form is known as spam. When individuals sign up for a number of websites, products, services, catalogues, newsletters, and other forms of electronic communication, they frequently receive unwanted spam and phishing communications, which, in some instances, may be potentially deadly [1; 8]. It has been discovered that viruses and Trojans can be used to generate spam email in certain circumstances. Recent findings from a survey that was just published by the China Anti-Spam Alliance reveal that the typical Internet user receives 35 emails each week, of which 41% are considered to be spam. Users of the internet and companies that provide internet services both find unwanted communication, sometimes known as spam, to be an irritant and a waste of time. In addition to this, they frequently distribute malicious software as well as content that is likely to be considered objectionable. As a direct consequence of these needs, cyber professionals have devoted a significant amount of time and effort to improving their capacity to recognize spam in digital communication. To combat spam, several strategies, including as blacklists, whitelists, decision trees, email addresses, and even machine learning, can be utilized. In addition, there is a broad spectrum of possible approaches. Examining the language contained inside the primary portion of an email is the primary method through which the great majority of them receive information. As a consequence of this, there is an increasing demand for efficient anti-spam filters that can instantly delete spam messages upon detection or inform users when they receive a message that may be spam. These filters are needed because: The problem with spammers is that they are always looking for new ways to get through anti-spam technology. They have come up with a brand-new method that will make the widespread distribution of spam email more simpler. They are in no way put at a disadvantage by the situation that currently exists as a direct result of this fact. To avoid

being identified as spam by filters, tokenization assaults will frequently make use of extra spaces. As a consequence of this, the information that is contained in emails needs to be arranged [14]. Despite the fact that machine learning-based spam email detection is currently the most accurate method [3; 17], it has a problem with false positives (precision of 92.9%) as a result of its inability to differentiate email threats on more than one occasion. This causes it to have a problem with false positives. A recent attack using the malware known as WannaCry is one example. We are able to clean up texts in preparation for future examination by using the method that we have outlined. This includes getting rid of stop words and other extraneous content that may lead to false positives or adjustments to the attack plan. After going through preprocessing, the texts are put through a number of different procedures that are aimed to pull out a range of characteristics. Word2vec, n-gram, character n-gram, and an n-gram combination with variable length are the components that make up these procedures. The classification of emails makes use of a number of different machine learning algorithms [15], such as support vector machine (SVM), decision tree (DT), logistic regression (LR), and multinomial naive bayes (MNB).

2.2 MALICIOUS LINK DETECTION

The authors Liu et al. proposed a content-based spam filter (2). Instruction and categorization are the two key foundations of their strategy for resolving the issue. When comparing the keywords extracted from individual users' emails to a corpus of spam and ham keywords, an overall accuracy of 92.8% and a precision of 84.6% were attained. This was determined by comparing the two items. Gaurav et al. suggested a system for identifying unsolicited email (3). In order to identify spam emails, this system classified emails using the document labeling idea and three algorithms (Multinomial naive bayes, Decision Tree, and Random Forest). Random Forest fared the best among these classifiers, with an accuracy and precision of 92.97% and 92.9%, respectively. Ioannis Kanar et al. proposed a cost-effective method for detecting spam that is database-based (4). In lieu of the 'bag of words' method for feature extraction, they constructed a 'bag of character n-grams' by generating the bag using character n-grams. The accuracy of the algorithm proposed by Kiliroor et al. (5) to identify spammy posts on social media profiles was 91.18 percent. Weimiao Feng and coworkers created an approach that employs the SVM-NB (6). The SVM approach is utilized for both the classification of training data and the identification of training samples that are dependent on other training samples. The authors Moon et al. (7) suggested using n-gram

indexing in tandem with support vector machines to combat spam emails. The team practiced the SVM classifier filtering procedure using actual user emails as training material. Kaur et al. suggested methods for the detection of spam based on N-gram analysis and machine learning (8). The produced N-grams are subsequently used for the prediction of unlabeled data. Ahmad et al. (9) proposed a method with an accuracy of 96% by combining a support vector classifier with a thoughtfully selected subset of variables for training purposes. Sarker et al. have exhaustively explored and studied the efficiency of machine learning security modeling that incorporates the best features (16). Nayak et al. (10) proposed a method for recognizing spam emails by employing a combination of the Naive Bayes and Decision Tree machine learning algorithms as classifiers and a hybrid bagging approach as a feature. This method employs a hybrid bagging strategy. As a result, the overall accuracy was 88.12%. Sheu et al. (11) presented a decision tree classifier as a technique of commencing a search for spam association rules by analyzing email headers. Next, we employ the association rules to develop an efficient and accurate approach of systematic filtering. Chen et al. (12) established a systematized spam filtering technique based on decision tree data mining methodology as a means of studying spam association rules and incorporating those analyses into the development of an efficient spam filtering method. [C]hen et al. This approach has a 96% chance of being effective. Kumar et al. (13) suggested a method that analyzes the email's header and URL in addition to doing multiple rule-based analyses on the message's text. In addition to the Apriori method and the Bayesian classifier, they were able to search through files and attachments to discover the relevant categories. The authors Khamis S.A. et al. (17) suggested a method for identifying spam that leverages information derived from email headers. By comparing two email databases, this methodology was created. A Support Vector Machine was applied to get an accuracy rate of 88.80 percent while classifying emails.

2.3 PHISHING DETECTION

2.3.1 Phishing on Social Networking Sites

In recent years, there has been a meteoric rise in the number of people using various social networking platforms. People communicate with their friends and trade various sorts of media with those friends. Sites that facilitate social networking, such as Facebook and Twitter, consistently place among the top 20 websites that receive the most traffic [36]. According to the research, the typical person devotes a much greater proportion of their time

to social networking websites than they do to any other kind of website. Because of the ever-increasing size of their user bases, social media websites are now in a position to compile significant data on their users' acquaintances, habits, and even economic circumstances. Through the use of social networks, dishonest people are able to communicate with anybody they like. In 2008, 83% of users reported receiving at least one friend request or communication from someone they did not know [37]. [S]ome users reported receiving multiple friend requests or messages. It is simple to mimic another user and obtain access to their trusted circle of friends on social networking sites because of the weak authentication procedures that many sites use [16]. As a direct response to the exponential growth of spam email, a wide variety of different phishing screening systems have been created. As illustrated in [17], targeted phishing is feasible because of the information that is provided through online social networks. [Citation needed] Regarding Twitter, a phishing simulation was carried out on that platform by [18]. Phishers started using the hashtag in their communications after it became popular on Twitter and became a trending topic. It also discusses features, such as node degree and message frequency, that may assist you in identifying a phisher and protecting yourself from falling prey to their tricks. Both the detection of phishing and the behaviour of users have been the subject of a significant amount of research. A Bayesian classification method is utilized in order to identify potentially suspicious behaviours from those that are more usual. The authors of [26] study a dataset as well as Twitter's performance, and they come to the conclusion that the Bayesian classifier is the method that is the most effective at identifying phishing. In today's world, it is common practice for client software and server software to both contain some kind of filtering mechanism as part of their functionality for managing incoming and outgoing email. It is recommended to use a Bayesian phishing filter rather than a static keyword-based phishing filter because the former can more effectively combat the latter. Poisoning Bayesian phishing filters can be done in a straightforward manner by avoiding known phishing keywords and replacing them in the emails with a large number of innocuous phrases. In addition, it takes them a considerable amount of time to adapt to a new method of phishing depending on the feedback provided by victims. In addition, very few of the currently available phishing filters make use of social networks as a tool to assist in the detection of phishing. When it comes to the shortcomings of the Bayesian phishing filter, there is an alternative: the friendly user interface of the Social Network Aided Phishing Defence

(Sniffer). We make use of a phishing filter that is both extremely effective and individualized (SOAP) [27]. SOAP exploits the social interaction that occurs between email correspondents to leverage the social relationship that exists among email correspondents in order to identify phishing. This allows for phishing to be detected in an adaptable and automatic manner. SOAP integrates three distinct components into a single whole in order to produce a Bayesian filter that is more resilient. Phishing filtering based on social interests, adaptive trust management, and phishing screening based on social proximity are all examples of this type of technology. According to the findings of many experiments, Simple Object Access Protocol (SOAP) has the potential to dramatically improve the accuracy, attack-resilience, and efficiency of Bayesian phishing filters.

2.3.2 Email Phishing

Email is the most popular form of online communication at this time. Over ninety percent of all email traffic is now composed of phishing emails, which has occurred concurrently with the parabolic rise of email. Therefore, over 120 billion of these types of communications are sent every single day [25]. Try sending out some mass e-mails if you want to communicate with a huge number of people while keeping your costs down [13]. From this point of view, phishing is a waste of resources due to the time that is wasted reading spam and the utilization of system resources (such as a computer's processing power and available disk space) to disseminate malicious programs. Those who engage in phishing and supply fake information can easily exploit the structure of email systems to their advantage. Every single email service available on the Internet makes use of a protocol known as "SMTP," which stands for "Simple Mail Transfer Protocol." SMTP was designed to keep a log of the path that an email takes as it is sent from the sender to the recipient. As a matter of fact, there is no privacy in email, it can be changed while it is in route, and the identity of the sender cannot be validated due to the lack of protection provided by the SMTP protocol. When a person receives an email message, they have no way of knowing who delivered the message or if any other users have also received it. Phishers are able to commit huge amounts of fraud over the Internet because of the security holes that are present in SMTP, specifically the inability to authenticate the identity of the sender. Strategies for Combating Phishing Attempts Via Email The novel hybrid model that was developed in is called Partitioned Logistic Regression, and it incorporates the most beneficial aspects of both naive Bayes and logistic regression. The original feature space is segmented into a large

number of separate feature classes using this modelling approach. It was suggested in [7] that users' anonymity may be preserved through the use of decentralized phishing screening. The system makes use of a structured peer-to-peer architecture across mail servers in order to communicate information regarding phishing in a cooperative manner. Additionally, the system makes use of powerful digests in order to identify messages that are only slightly different from one another. An ant-colony based phishing filter was developed by [21] so that it could evaluate and anticipate phishing communications. Several distinct approaches, including multi-layer perception, naive Bayes, and the Ripper classifier, are contrasted with the phishing filter that has been created. Phishing assaults may be anticipated in a more accurate manner thanks to the newly developed filter, which also offers an extra technique of doing so. [22] Created a rule-based filter that is both lightweight and accurate in its detection of phishing attempts made over email. This filter makes use of a cascading system that consists of three filters: one for analyzing the fingerprints of message contents, another for comparing the sender's address to a white and black list, and a third for identifying phishing and legitimate email-related keywords in the header of the email. When employing this method, you will be able to send an incredible 90 emails each second.

2.3.3 Image-Based Phishing Attacks

Phishers have been using a technique known as "picture phishing" more frequently in recent years. This technique entails embedding a phishing message into an image that a target may view rather than within the message body itself. Phishing emails typically have images attached to them, which conceal the fraudulent information. The goal is to circumvent the possibility of phishing filters, such as automatic text classifiers, reading and analyzing the contents of the textual portion of the email. Because the vast majority of email clients automatically load any attached pictures, the receiver is able to immediately understand the message that was intended for them. The most fundamental method of image phishing is simply taking a screenshot of some plain text that was written in a standard text editor.

An illustration of an image phishing scheme can be shown in Figure 2.

Traditional content filters are not very good at detecting phishing attempts that use pictures. In order to filter these signals, creative strategies are required. Images that are used in phishing attacks are frequently made by making arbitrary changes to a stock photo that serves as the basis for the assault. Obfuscation is sometimes used to prevent optical character

recognition (OCR) software from interpreting hidden text because this renders signature-based detection approaches worthless. It is quite humorous that the same text obfuscation techniques that are used to trick OCR systems are also used in CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart). The user is shown an image that has been distorted, and they are required to write down the letters and/or numbers that match to the image. This is one variation of the conventional CAPTCHA. Due to the fact that a typical human being is able to read a CAPTCHA without any issues, but an internet bot is unable to interpret the picture letters, such tests are widely utilized in order to prevent unwanted internet bots from accessing websites. [19] Describes a method for quickly filtering phished photographs that utilizes both server-side cluster analysis of phishing photos as well as user-side active learning classification of the results. Extensive experimental evaluations of server-side and client-side algorithms were performed on a real-world image phishing dataset received from a mail server. These evaluations were used to demonstrate that this strategy is effective. [20] Propose a method that uses machine learning to differentiate between the photos used in phishing emails and those used in real emails. Using the recommended strategy, effective global picture attributes are retrieved, which are then utilized to train a sophisticated binary classifier to recognize phishing images. Even with such a limited sample size, the preliminary findings have been very positive. The proposed method achieves a level of performance that is deemed satisfactory when it is tested using fivefold crossvalidation on a dataset that contains 928 phishing pictures and 810 nature shots.

2.3.4 Click Baits

Phishers create false clicks and route traffic to their websites in this stage of the attack in order to win control. Phishers will employ search engines to conduct their activities, submitting queries and following connections to reach their targets [28, 34]. Phishers can also be motivated to manufacture phony clicks by the incentives offered by internet advertising [30].

2.3.5 Sophisticated Phishing Disguised as Content

In order to participate in content phishing, an individual must first change the search engine's reasonable interpretation of the contents of the page. Content stuffing is a type of content phishing in which keywords are intentionally placed within a webpage in order to artificially

inflate the page's keyword count. This is done in an attempt to rank higher in search engine results for those keywords. Figure 3 depicts the web page that can be reached by entering "video songs hd" into Google's search bar and then selecting one of the links that appears as a result of that search. Phishing attacks that target content can take a variety of forms, including those that target the title tag, the body tag, or the Meta tag. The development of link-based ranking algorithms [5, 68] has helped to alleviate some of the problems caused by the content phishing phenomena. But because phishing is a constantly developing technique, phishers rapidly started building link farms [31, 32]. Problems associated with the identification of phishing have previously been resolved thanks to the application of data mining techniques [38]. The purpose of content-based phishing detection systems is to automatically recognize a message sent by a phisher as phishing. This is accomplished by doing an analysis of the content of the message and locating patterns that are indicative of phishing. [39] We present a new approach to content-based phishing detection that works by first discovering hidden patterns in both phishing and valid messages, and then allocating each pattern to the class that is most applicable to it. In other words, the method works like this: For the purpose of message classification, it makes use of both linguistic and symbolic techniques. This method allows for the classification of more than one hundred messages every second, which contributes to the system's high level of computational efficiency.

2.3.6 Covert Operations and Route Alteration

In search engine optimization (SEO), a cloak is a strategy for concealing information from both the user and the search engine crawler. It is possible to immediately send a user to a different URL by using redirection after the user has loaded the required current URL. Cloaking and redirection are two techniques that are utilized in search engine phishing [33]. Phishers often look in the user: field of an HTTP request and keep track of the IP address that is used by search engine spiders to differentiate between actual users and bots. Utilizing Java script that is activated either by the OnLoad() event of the page or by a timer is yet another approach that may be utilized to direct viewers to malicious websites. Because the most majority of web crawlers are script agnostic, it is unlikely that they will be able to detect the thing that must be noticed in Java script, which is the most prevalent and widespread type of script. There are very few strategies for identifying cloaking; however, one such method is described in [8], and it makes use of the most popular terms from the MSN inquiry log in conjunction with the words that generate the most money from the MSN

advertisement log. Cloaking might potentially be found through the process of comparing crawls carried out by robots with different user agent strings and IP addresses. Phishers, on the other hand, are adept at coping with robot behaviour, collecting and disseminating the IP addresses of crawlers, and successfully differentiating automated browsers from human ones. Phishing can also get around Web phishing filters if the filters rely on information that is out of date, which can trick web crawlers into thinking that phishing websites are authentic. The widespread practice of employing parking domains for phishing campaigns is a crystal obvious illustration of this principle in action. Phishers will often buy up defunct genuine businesses' domains and fill them with malicious content after the businesses go out of operation. These websites continue to show with the same content for some time after they have been deleted from the index and the input for the phishing classificatory.

2.3.7 Phishing through URLS

Phishing that is accomplished through the use of a hyperlink is also referred to as comment phishing and blog phishing. This is a sort of phishing that exploits social media platforms including blogs, guestbooks, message boards, and wikis to deceive users into giving out personal information (wiki phishing) (wiki phishing). The target might be any website function that shows user-submitted links or tracks where users come from by tracking their referring URLs. To boost their search engine results, phishers will often spam online guestbooks with links to their own sites, which are often completely irrelevant to the topic at hand. When it comes to phishing via links, there are two main categories: outbound and inbound. Phishing via outbound links is the simplest and most cost-effective method. In this case, the server has unrestricted access to his sites and can thus add anything to them and generate an extensive network of reliable references. When attempting an incoming link phishing attack, the phisher simply boosts the number of links leading to his intended victim. Setting the page's refresh time to zero and populating the refresh URL property with the URL of the target page is the quickest approach to trigger a redirect. More effective from the perspective of the phisher is the deployment of page-level scripts, which are typically not run by crawlers. Trust [9], mistrust [10] propagation in the neighbourhood, or their combination [35], and graph-based similarity measures [11] are just a few of the recent results that employ rank propagation to generalize trust or distrust judgments made across a small group of seed pages or sites to the entire web. Researchers have lately proven success in recognizing link phishing by utilizing temporal data and genetic programming factors.

3. THEORETICAL FRAMEWORK

In this chapter, the theory on which this study is based is presented.

3.1 E-MAIL

Electronic mail or e-mail was created even before the Internet. The first standards were proposed in 1973 with RFC 561. With the conversion of the ARPANET to the Internet in the early 1980s, e-mail as it is known today began to take shape.

Currently, the format of e-mail messages is defined by RFC 53222. This RFC defines the division of e-mail into two main parts:

A. header: first part of the email, organized into fields.

B. body: part of the email that contains the text of the message.

The e-mail messages are formed only by texts. Such texts use characters, interpreted as ASCII, with values in the range 0 to 127. The texts are divided into lines. The end of each line is marked by a sequence of two characters: carriage return (CR) and line feed (LF), referred to by the acronym CRLF in the specifications and also throughout this work.

The header is separated from the message body using a blank line, that is, a line that contains only the CRLF characters.

Lines of text in a message cannot exceed 998 characters and it is recommended not to exceed 78 characters. The 988-character limit was created in order to maintain compatibility with various implementations that were not designed to handle lines with more than 1000 characters. The recommendation to keep lines up to 78 characters long was created so that the interface of several applications would not be harmed, as many applications display a maximum of 80 characters per line.

3.1.1 Email Header

The header of an e-mail consists of lines of text. Each row contains a field. Thus, each line in the header starts with the name of a field, followed by the character “:” (colon), the field body and the CRLF pair. The field name must consist only of printable characters from the US-ASCII standard, more precisely, the characters between the values 33 to 126, except for the comma. The body of a field can consist of the same characters, in addition to white space and vertical tabs. In figure 3.1 it is possible to observe some lines of headers and their format.

```
From: John Doe <jdoe@machine.example>
To: Mary Smith <mary@example.net>
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.machine.example>

This is a message just to say hello.
So, "Hello".
```

Figure 3.1: Content of an email: its header and body of the message

There are also rules for unstructured (unstructured) and structured fields. Unstructured fields are treated as lines of characters only. Structured fields have semantic rules defined and a list of appropriate tokens. Rules for creating long lines of text in headers are also defined, which allow you to split a long header field into multiple lines. The fields of a header carry information, such as sender, recipient, sent date, subject, date, among others. The only mandatory ones are the date of origin and sender.

3.1.2 Email Body

The message body is simply a sequence of lines of text using only US-ASCII characters. There are only two rules for the message body:

A. The characters CR and LF can only occur together, forming the pair CRLF.

B. Lines of text are limited to 998 characters, although it is recommended that they be limited to 78 characters, not counting the CRLF pair.

3.1.3 Non-Text Content

As described earlier, the content of an email is restricted to American ASCII characters only. This fact greatly limits the content of an email message, due to the absence of appropriate characters used in other languages. In addition to being limited in relation to the various existing languages, the impossibility of transmitting information other than text greatly restricts this means of communication.

To circumvent this limitation and still maintain compatibility with existing systems, the Multipurpose Internet Mail Extensions, better known by the acronym MIME, was proposed and adopted. MIME is defined by five main RFCs:

a. RFC 2045: defines the various headers used in MIME messages.
b. RFC 2046: defines supported media types.

c. RFC 2047: describes extensions that allow the use of non-US ASCII characters in header fields.
d. RFC 4289: defines procedures for registering new transmission encodings.
e. RFC 6838: defines methods for registering new media types.

In addition to the aforementioned RFCs, there are others that update or correct these and others that became obsolete after the creation of the new RFCs. For example, RFC 6838 made RFC 4288 obsolete, which had, in turn, made RFC 2048 obsolete.

In summary, it can be said that MIME adds three main functionalities to the original email system:

f. Definition of different types of content (content type).
g. Definition of transmission encodings, which are used to represent 8-bit binary data using only the 7-bit characters of the ASCII character set.
h. Rules for encoding non-ASCII characters in message header fields.

In addition to providing new features, MIME maintains compatibility with the original systems. This compatibility allows, for example, that a MIME email client application can correctly interpret a non-MIME message, as well as allowing a MIME message containing texts to be correctly processed by a non-MIME client. Support the MIME standard.

In order to implement the aforementioned functionalities and maintain compatibility with the original e-mail system, a MIME message is distinguished from the others by the header field “MIME-Version: 1.0”. This field indicates that the email contains MIME version 1.0 content, a version that has not been changed so far.

After the header, a MIME message can contain the “common” content of an e-mail, that is, a body of text formed by only ASCII characters, according to the rules described above. This body is included even when the message has text in other formats, such as HTML. This allows email clients that do not recognize the MIME standard to also display a readable message to their users.

The Content-Type field is also included in the header of the MIME message, which defines the type of content of the message. This field is optional and, when it is not included in a MIME message, it means that the message contains only plain text, that is, only ASCII characters.

The Content-Type header field is formed by a type and subtype, for example:

i. Content-Type: text/plain: sets the type/subtype to plain text.

ii. Content-Type: image/jpeg: defines the type/subtype as an image in jpeg format

New types of content can be created. To be used, they must be registered according to the standard defined by RFC 4289. Registration is not required to make it difficult to create new types, but to avoid possible name collisions.

If the content to be transmitted in a MIME message cannot be represented by ASCII characters, this content is then encoded by some method, such as, for example, the base64 method. Supported encoding methods define schemes for encoding data using only ASCII characters supported by the original email system. In this way, full compatibility with existing systems is maintained. For example, to indicate that the e-mail content is encoded in base64 format, the header field "Content-Transfer-Encoding: base64" is used.

The MIME standard also allows, through RFC 2045, the inclusion of new encoding methods, which employ private formats and rules that indicate how these should be included and differentiated. of standardized methods. However, this same RFC discourages its use.

iii. multipart Messages

The MIME pattern described so far solves the problem of transmitting information that could not be represented by the ASCII character set. In addition, the pattern also allows a message to be formed not only by one content, but by multiple contents, where each of these contents is found within a entity.

In messages with multiple contents, it is possible to transmit, for example, textual content and several images, which can either be in files attached to the e-mail or incorporated directly into the text. For example, the body of the message, in HTML format, can contain an entity with the HTML text and several entities with the images and other components referenced by the HTML content.

A multipart message follows the same rules, already described, as any MIME message. The only difference is in the way your content is interpreted. In addition to the MIME-Version header field, which indicates that the message follows a MIME format, the Content-Type field must contain the value multipart/mixed, which indicates that the message has many parts. The mixed subtype indicates that different contents are present in the message.

In the mixed subtype, the order of entities is important and must be respected by email clients. Thus, whenever a message with type multipart/mixed is displayed, the client must display each entity in the order in which it is found in the message.

In addition to the mixed subtype, multipart entities can have the values alternative, digest, and parallel. In the first case, the subtype indicates that all the contents of the entities are the same, differing only in the way they are encoded. This subtype can be used to provide an email client with the same message in different formats, leaving the client to display the message that is most convenient for the user. The second subtype, digest, is used to send collections of messages. It is semantically identical to mixed, differing only in the content assumed by default when this is not specified. Finally, the parallel subtype is similar to the mixed one, differing only in that the email client no longer needs to respect the order of the entities.

To allow multiple entities to be handled properly, they must be separated by a value that delimits them. Thus, whenever a multipart Content-type header field is specified, in addition to the content type, it is necessary to specify the boundary value between the entities. The boundary value specifies the start and end of each entity, as can be seen in Figure 3.3.

Entity delimiters are generated by the agent that composes the email message and it is your responsibility to ensure that the delimiter value is unique in the message and appears only as a delimiter. When used to separate entities, the delimiter value is preceded by two hyphens. For example, the delimiter in Figure 3.3 is the text “simple boundary” and therefore in the message body it is always displayed as “–simple boundary” (without quotes).

A multipart message also allows each entity to be multipart, forming a tree-like structure. Thus, it is possible to create emails with very complex contents. The delimiter of each entity must, however, remain unique throughout the email.

```
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Date: Sun, 21 Mar 1993 23:56:48 -0800 (PST)
Subject: Sample message
MIME-Version: 1.0
Content-type: multipart/mixed; boundary="simple boundary"

This is the preamble.  It is to be ignored, though it
is a handy place for composition agents to include an
explanatory note to non-MIME conformant readers.

--simple boundary

This is implicitly typed plain US-ASCII text.
It does NOT end with a linebreak.
--simple boundary
Content-type: text/plain; charset=us-ascii

This is explicitly typed plain US-ASCII text.
It DOES end with a linebreak.

--simple boundary--

This is the epilogue.  It is also to be ignored.
```

Figure 3.2: Content of a multipart email, showing its headers and message body with two textual contents

3.1.4 E-Mail Transmission

E-mails are transmitted using the Simple Mail Transfer Protocol (SMTP), defined by RFC 5321. Users generally use an application called a client, such as Outlook Express, for example. Compose and convey your messages. The client makes use of the SMTP protocol to deliver the message to an e-mail server, commonly the server where the user has his mailbox. This server, upon receiving the message from its user, uses the protocol to transmit the message to the server (or servers) to which the message is sent. The SMTP protocol is only responsible for the transmission of e-mails. These are stored in mailboxes until they are "recovered" by the user. To access your messages, the recipient uses a client, which accesses the mailbox and retrieves the messages, using the Post Office Protocol (POP), defined by RFC 1939, or the Internet Message Access Protocol (IMAP), defined by the RFC 5301.

a. SMTP Protocol

The SMTP protocol is used to transport e-mails. The protocol defines commands in text format, using ASCII characters, and is, in large part, responsible for the limitations of the

format of e-mail messages, such as, for example, the limited number of characters per message line. An SMTP server acts both as a receiver of messages, when it is the recipient of an email, and as a transmitter. The server acts as a transmitter in cases where the received message does not have it as a recipient. In this case, the server is responsible for transmitting the message to the destination server, or to another transmitting server, which will be responsible for forwarding the message to its recipient, or another transmitter. The process repeats until the message reaches its destination. One of the major problems of this protocol is the absence of strong authentication mechanisms. This allows malicious users to falsify the origin of messages, which is one of the common techniques for generating spam, as described by Stern

b. Email submission

With the constant increase of malicious devices on the network and, consequently, of worms, spam, viruses, and other forms of malicious content, many servers have prohibited the sending of emails through port 25 (default port). to that of the SMTP protocol). With the increase of these malicious devices, servers sending messages are currently required to be responsible for the traffic they generate, requiring them to send messages through an authentication process. Thus, as a consequence, the process of submitting e-mails to servers is separated from the process of transmitting e-mails between servers. This separation between submission and transmission of e-mails is defined by RFC 6409. The RFC defines that, before forwarding a message, any server must make sure that all domains found in the message are qualified. If they are not, it must be rejected. The server must also require that the connection through which it received the message to be forwarded be authenticated. The RFC defines forms of authentication. In addition, the RFC also suggests that the server check the syntax of message addresses, log client configuration errors, and have less tolerance for timeouts.

c. POP protocol

The Post Office Protocol (POP) was developed to allow a user to access their mailbox (mailbox) on an email server. Thus, an SMTP server takes care of receiving messages uninterruptedly and the POP protocol provides mechanisms for the user to transfer, whenever convenient, the e-mails from his mailbox to his workstation. POP was not designed to handle large volumes of messages. Normally it's just used to transfer your

messages from the server to your workstation. After transfer, your messages are removed from the server.

Like the SMTP protocol, the POP protocol is based on textual commands, composed of characters from the ASCII table and terminated by the CRLF pair. Its functioning can be represented by a state machine, where the initial state, which starts as soon as the connection is opened, is called authentication. In this state, the server sends a “greeting” to the client and then waits for its authentication. Upon receiving a valid authentication, the server allows exclusive access to the requested mailbox and transitions to the transaction state.

In the transaction state, the client can request information about messages on the server, request their transfer and request that they be removed. It is important to note that when a client requests the removal of a message, the server just marks the message as removed. The removal itself occurs only when the client normally closes the connection.

When the client finally sends the “QUIT” command, the server transitions to the update state. In this state, the server effectively removes the messages and closes the access to the mailbox. If the connection is closed by any method other than the “QUIT” command, the server does not proceed with the removal of the messages.

d. IMAP protocol

The Internet Message Access Protocol (IMAP) is a protocol for accessing mailboxes on servers, such as the POP protocol. Unlike this one, however, the IMAP protocol removes messages from the server only when the user explicitly requests it.

Furthermore, while a POP client copies all messages from the server to the user's workstation, an IMAP client only copies messages that the user requests, which is advantageous in this case. Of users who receive many messages. Yet another difference in relation to the POP protocol is that IMAP allows several clients to be connected and concurrently accessing the same mailbox. The IMAP protocol also makes it possible, in the case of MIME messages, for the user to access only the parts of the message he wants. For example, when receiving a message with attached files, the user can only access the text content, not needing to copy the entire message to his workstation, saving time and computational resources. IMAP also makes it possible for messages on the server to have status information, such as whether they have been read, replied to or deleted. The status of each email is generally maintained by email clients. Storing the status of each e-mail on the server allows different e-mail clients to make use of and take advantage of this information.

Users of the IMAP protocol also have the possibility to create multiple mailboxes on the server, rename them or remove them easily, as they are presented to the user as if they were folders. In addition to manipulating their mailboxes, it is possible for the user to transfer messages between them. In addition to these facilities, the IMAP protocol also offers a system with several search criteria. It also allows the creation of extensions to the protocol itself. However, there are also disadvantages of this protocol in relation to the POP protocol. For example, the more complex computational implementation, greater consumption of server computational resources, among others. Like the SMTP and POP protocols, the IMAP protocol is based on textual commands, composed of characters from the ASCII table and terminated by the CRLF pair.

3.1.5 Transmission of Images in Emails

The transmission of images in e-mails can be performed in different ways, due to the variety of resources offered by the MIME standard. Usually, however, images are sent in attached files or included directly in the body of messages. Regardless of the form used to transport images, this work assumes that a usual anti-spam system, used for the detection of textual spam, identifies e-mails that contain images. Thus, it sends them to a system specialized in recognition of spam images for analysis. The result of the analysis is then sent to the original classifier system, which makes the appropriate decisions about what to do with the message. Thus, the specialized system for recognizing spam images proposed here acts as an application that extends the functionality of existing textual anti-spam systems. Thus, this work aims to analyze only the issue of image classification, not addressing the integration of this classifier to the various existing textual anti-spam systems.

3.2 TYPES OF SPAM IMAGES

With the increasing accuracy of spam detection techniques aimed at the textual analysis of e-mails, spammers began to use images to transport their information and thus avoid textual filters existing. Spam images have various advertising content, as described below
Adult: images offering services to an adult audience, such as pornography, dating services, personal ads, among others; Financial: images with offers for financial services, stock market, investment opportunities. Such images may be included, along with fraudulent messages, in emails that aim to obtain passwords or banking information from their recipients; products: images that offer the most varied products and services. Internet:

images that offer computing or Internet services, such as, for example, domain registration, hosting and marketing services; leisure: offers of prizes, trips, casinos, among others; health: offers of food supplements, sports products, pharmaceuticals (Figure 3.3), of products for disease prevention, medical services, among others. Political: political advertisements, of products related to political parties, requests for financial donations to politicians and parties; education: course offerings in private educational institutions; religion: propaganda of religious services, of religious entities, evangelization messages.

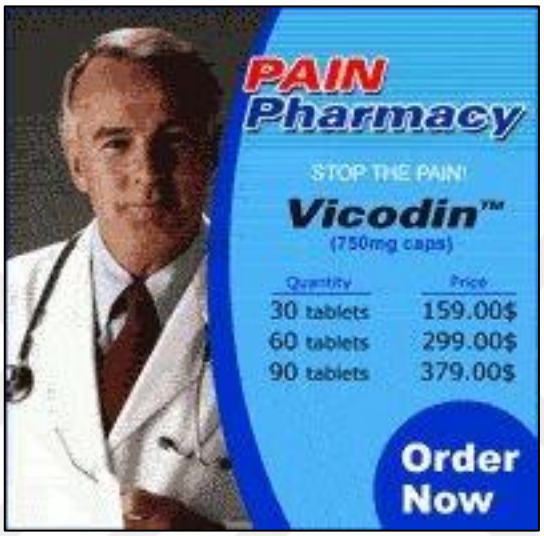


Figure 3.3: Example of A Spam Image Offering a Pharmaceutical Product

3.2.1 Tricks Used in Spam Images

Spammers use the most varied techniques to confuse the spam image detection systems. Some of these techniques are described below

a. Obfuscation

Obfuscation is one of the oldest and most well-known techniques. It consists of using misspelled words, slightly rotating texts, adding shading, hiding borders and adding random noises. In Figure 3.5, an example image using this technique is shown.

b. Textual Content

This technique employs images composed only of text, as, for example, in Figure 3.6. Despite its high computational cost, OCR techniques can be used to recognize spam in these images.

Figure 3. 4: Example of spam image with obfuscation

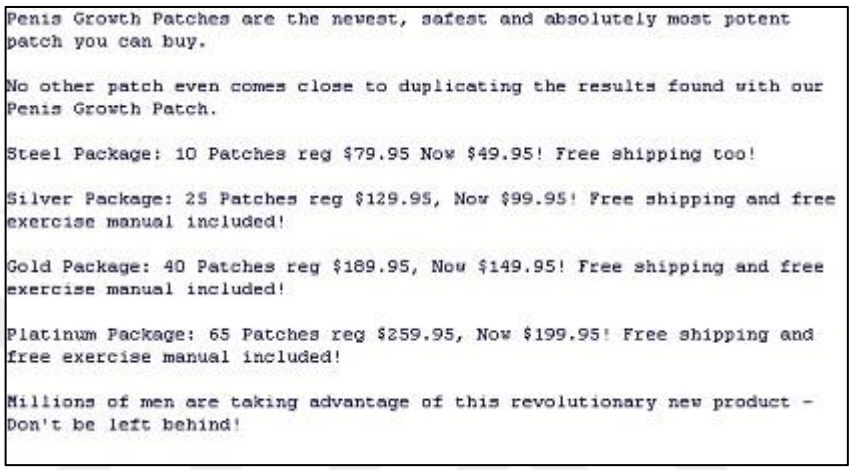


Figure 3. 5: Example of text-only spam image

c. Multiple Sections

In this technique, the image is divided into multiple sections, which makes its analysis difficult. The email client, however, reassembles the sections and displays the image correctly. The image can be randomly sliced. In Figure 3.6, an example of the technique is shown.

Figure 3.6: Example of spam image divided into multiple sections

d. Images with cursive writing

The use of cursive writing, as can be seen in Figure 3.7, is a technique used to confuse OCR-based systems, as cursive writing is not similar to any commonly used font.

Figure 3.7: Example of a spam image with handwriting

e. Background "Wild"

This technique consists of adding vague and strange backgrounds to the images. Geometric figures and a wide range of colors are used. The technique is used to confuse OCR techniques, as they look for letter-like geometric patterns in images. Figure 3.9 is an example of spam using this technique.

Figure 3.8: Example of spam image using “wild” background

f. Geometric Variation

This technique consists of altering the image only, but not the message it contains. Figure 3.9 presents an example of this technique.

Figure 3.9: Example of spam image with geometric variation. The messages are identical but use different backgrounds and sizes

g. gif images animated

The GIF (Graphics Interchange Format) format is widely used on the Internet, both for encoding still images and for displaying animated images. Animation is used by spammers to make it difficult to analyze and classify images. Figure 3.10 presents an example of an animated image.

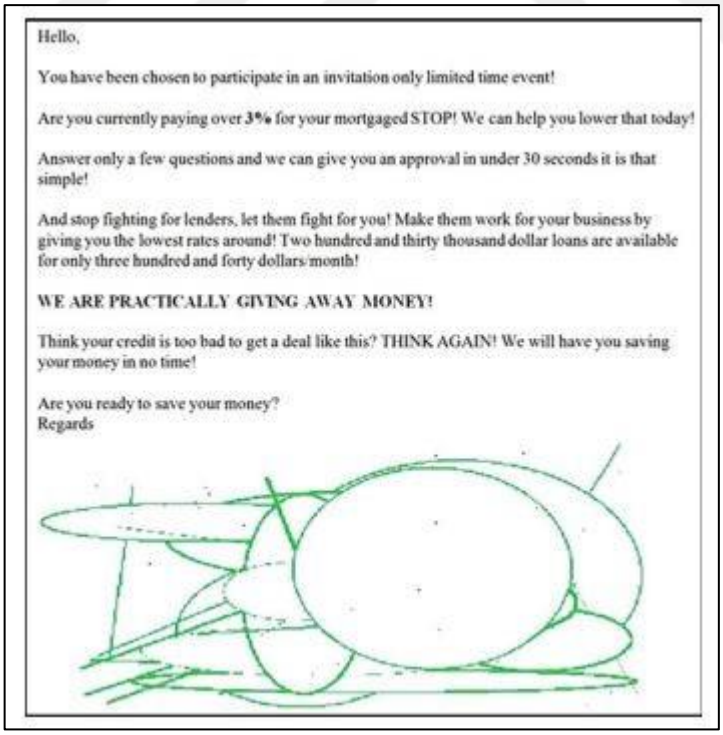


Figure 3.10: Example of a spam image using an animated GIF

h. Cartoon or Cartoon

This technique consists of using cartoon-style fonts, as shown in Figure 3.11. This font style produces a beautiful and attractive image for humans but difficult to parse for computer systems.

Figure 3.11: Example of spam image using image with "cartoon" style fonts

i. recolour

This technique consists of varying the attributes of an image, such as, for example, its colors. This technique can evade many image analysis techniques, such as histogram, pixel, and average color analysis. Figure 3.12 presents an example of this technique.

Figure 3.12: Example of two identical spam images, but using different colors

3.3 FEATURE EXTRACTION

For the extraction of features, an application was developed, in Java language, which processes groups of images, generating a file, in CSV format, with the characteristics of each image. In this work, the application was used on each of the image bases to generate the corresponding files for each base. A CSV file consists of fields separated by commas. Each line in the file represents a record and, optionally, the file can have a header. If it exists, the header, shown on the first line of the file, contains the names of each of the fields. The developed application generates the CSV files always in the same format. The number of fields, however, is variable, as it depends on the amount of characteristics that each digital image processing method generates. The first field is the name of the file that contains the image. After this field, there is a field for each value of each characteristic. Finally, the last field of each record contains the class — ham or spam — of the image. The inclusion of the field with the image class is necessary both for the training of the MLP neural network and for the evaluation of the classifications performed by it.

Figure 3.13 presents an example of a CSV file generated by the developed application. The first line contains the header, indicating the name of each field. Then each line represents an image. The first column contains the name of the file that contains the image. The second, third, and fourth columns contain the values of three characteristics — the average values of the RGB color channels. Finally, the last column indicates the image class.

```
filename,R,G,B,class
trec\ham\inmail.10293.0.png,0.15459333333333247,0.12501764705882298,0.13325647058823475,HAM
trec\ham\inmail.10293.1.png,0.0,0.0,0.0,HAM
trec\ham\inmail.10293.2.png,0.800000000000127,0.20000000000003176,0.4000000000000635,HAM
trec\ham\inmail.10293.3.png,0.8160552941177726,0.43549529411761734,0.5661705882353137,HAM
trec\ham\inmail.10293.4.png,0.35786352941175587,0.31822705882351976,0.26909490196076546,HAM
trec\ham\inmail.10293.5.png,0.9701960784313914,0.868365098039218,0.902205098039266,HAM
trec\ham\inmail.10293.6.png,0.5197768627450957,0.45255019607840696,0.46885333333330925,HAM
trec\ham\inmail.10293.7.png,0.4605996078431324,0.3385196078431161,0.25920941176469564,HAM
trec\ham\inmail.10293.8.png,0.9931447058823535,0.9912772549019612,0.9919019607843136,HAM
trec\ham\inmail.10295.0.png,0.7977776470588833,0.8204454901961219,0.8553376470588457,HAM
trec\ham\inmail.10295.1.png,0.9250674509804488,0.9487474509804342,0.9728427450980612,HAM
trec\ham\inmail.10295.2.png,0.8309996078431855,0.8392588235294594,0.8516301960784581,HAM
trec\ham\inmail.1031.0.png,0.09748823529411722,0.09817529411764674,0.11080078431372381,HAM
```

Figure 3.13: Excerpt from a CSV file used during this study

The application extracts the characteristics of the images according to each of the digital image processing methods listed below.

a. Average Color: calculation of the average values of the RGB channels of the images.
b. Histogram: generation of grayscale histogram of images.

c. Color Histogram: generation of the colored histogram of the images.
d. Color Moment: calculation of the color moments of the images.
e. Color Coherence Vector: generation of the color coherence vector of the images.

The developed application makes use of some functions of the ImageJ library to extract the characteristics of the images.

3.4 ARTIFICIAL NEURAL NETWORKS

An Artificial Neural Network is a mathematical model whose behavior seeks to simulate the operation of biological neurons. The Artificial Neural Network is composed of a set of artificial neurons, interconnected in a similar way to the interconnection of neurons in a nervous system. Artificial Neural Networks originated in 1943, when Warren McCulloch and Walter Pitts published a work on artificial neurons. A biological neuron has a cell body, called Soma. From the Soma, several filaments, called dendrites, and a long structure, called an axon, arise. At the end of the axon, there are connections to the dendrites of other neurons. These connections are called synapses. Through synapses, the axon of a neuron sends electrical signals to other neurons. Figure 3.14 presents a simplified model of a neuron. Signals are transmitted from one neuron to another through physicochemical reactions. Chemical transmitter substances are released at synapses and enter the dendrites, increasing or decreasing the electrical potential of the cell body. When the electric potential reaches a threshold value, the neuron fires an electric pulse through its axon. This pulse reaches other synapses and, through them, other neurons. Synapses undergo state changes in response to patterns of received stimuli. They are believed to form the basis for the learning of a brain

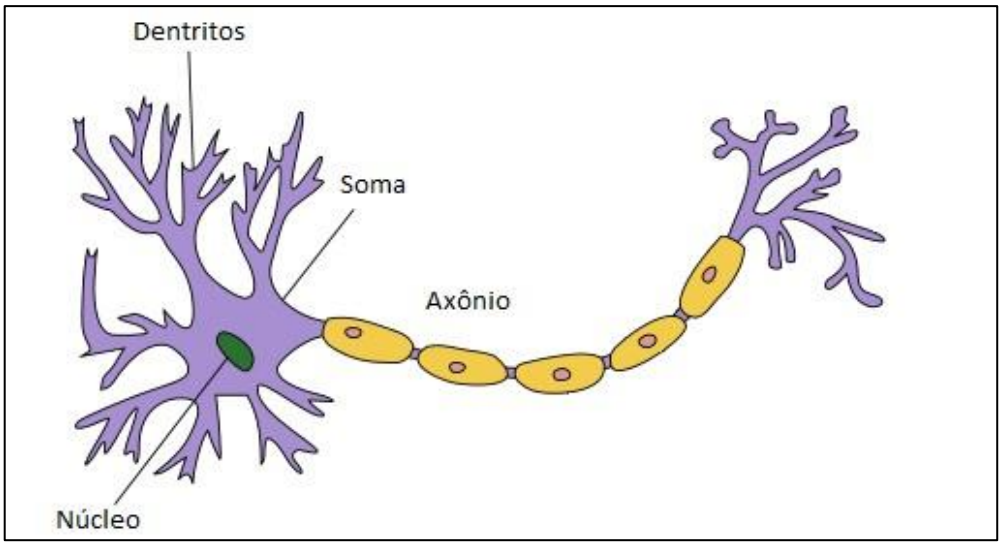


Figure 3.14: Simplified model of a neuron

3.4.1 Artificial Neuron

In an Artificial Neural Network, the fundamental processing unit is the artificial neuron, also known as a node. Nodes are connected to each other through weighted connections. The learning of an Artificial Neural Network is performed by adjusting the weights of these connections. Each node has an activation level, input connections and output connections. Through its input connections, a node receives the activation levels of other nodes and, from its output connections, it transmits its activation level. Dog to other nodes. Each node computes its activation level, based on the connections with the axons of its neighbours, without the need for a global control over the entire set of nodes in the network Figure 3.15 shows the structure of an artificial neuron.

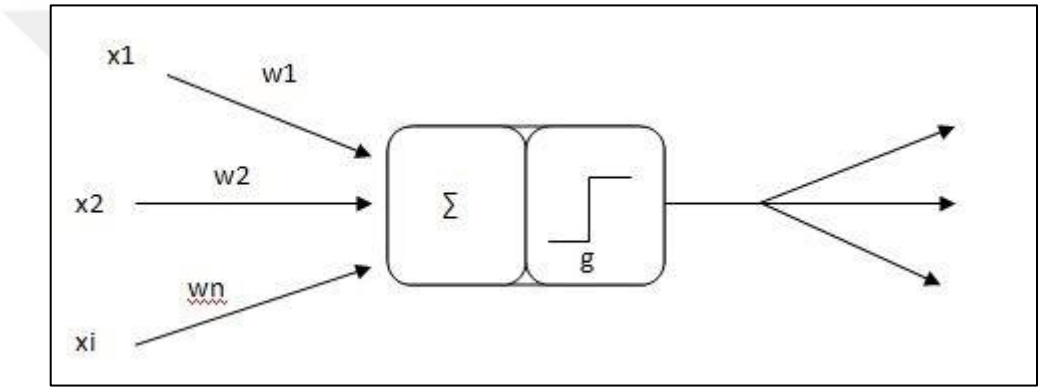


Figure 3.15: Artificial Neuron

The activation level ak of a node is given by

$$ok = g(\sum_{i=1}^{no} xiwki + bk) \qquad (3.1)$$

Where g is the activation function, xi is the activation level received from input node i, wki is the weight of the connection between the nodes iek, and bk, the bias of node k.

Different activation functions can be used on network nodes. The main ones are the step, linear, sigmoid and hyperbolic tangent functions.

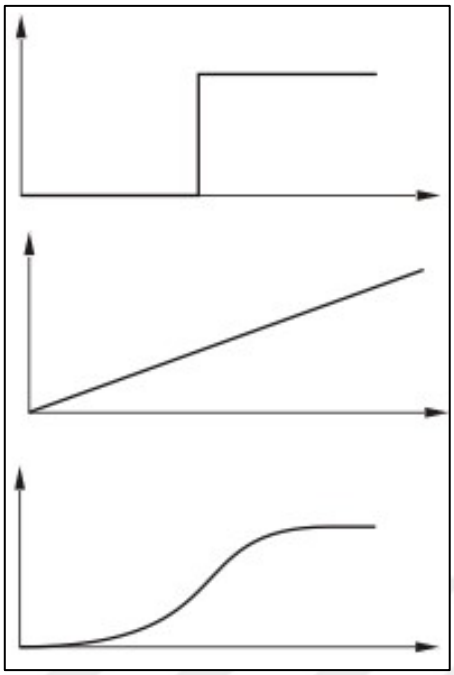


Figure 3.16: Graph of the step function, Graph of the linear function. And Graph of the sigmoid function

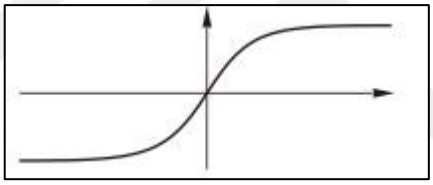


Figure 3.17: Graph of the hyperbolic tangent function

3.4.2 MLP Networks

Multi-layer-perceptron (MLP) networks are Artificial Neural Networks that have neurons arranged in layers — an input layer, one or more intermediate layers, called hidden layers, and an output layer. Figure 3.18 presents the architecture of an MLP.

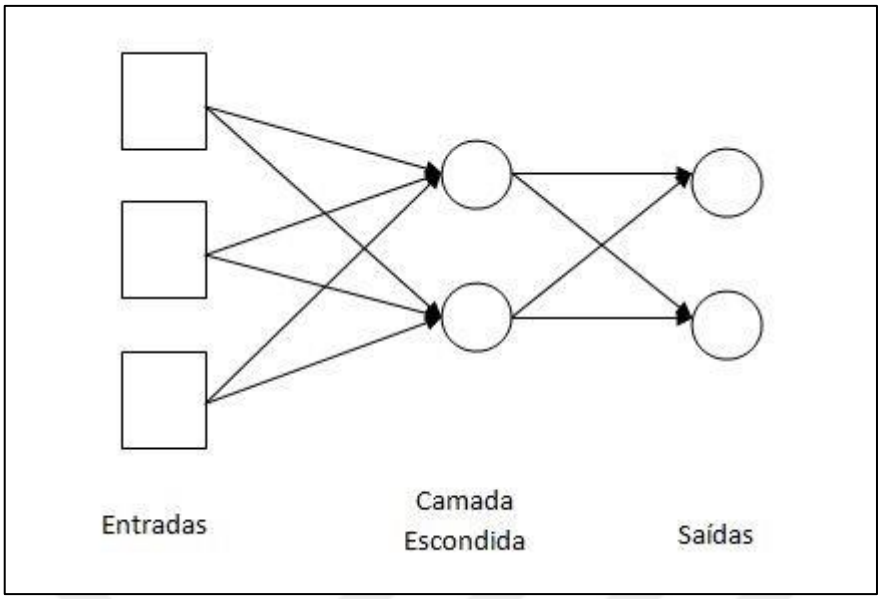


Figure 3.18: MLP network architecture, with a hidden layer

Back-propagation it is the most popular algorithm used for training an MLP. The training is done in a supervised way. The algorithm has two steps — one forward and one backward. In the first step, the MLP network is presented with a training pattern. The output produced by the network is then compared to the expected output for that pattern. In the second step, the network error is calculated. The error is passed retroactively to the correction of the weights of the network connections. The algorithm is called back-propagation due to the fact that, during training, the correction of weights occurs retroactively, that is, from the output neurons to the input neurons.

4. METHODOLOGY

4.1 INTRODUCTION

Phishing is a chronic problem to obtain personal information like account number, password, Social Security Number, etc.… through internet. Initially phishing is done using AOL tool where the hackers tried to snoop America online user's personal information. In the mid of 2016 the number of phishing websites around the world was approx. 100000 so, companies like pay-pal, G-Mail, etc. have worked on Anti-Phishing tools like secured servers, https, etc. to stop luring of personal information [2]. This chapter contains the technique to find whether the website in the internet is phishing or non-phishing based on certain features. The major problem here is due to these kinds of phishing websites many thefts occur in day-to day life, to stop this we need to completely ban these phishing websites. So, the first step in solving the problem is identifying the website based on features like URL length, Prefix suffix, SSL Service length, etc Online car ordering is an inexpensive and environmentally friendly means of transport, supply sharing is a brilliant idea that provides people with the convenience of renting cars in various easily accessible locations, and also offers another short-distance transportation option that allows them to travel without worrying about getting stuck in traffic, especially in busy areas like the city center. Supply sharing systems are an automatic way to rent cars. Currently, there are around 500 supply-sharing programs around the world. The characteristics of the data generated by these systems make them attractive for research. Unlike other transport services such as bus or metro, journey time, start and end position are explicitly recorded in these systems. This feature turns the supply sharing system into a network of virtual sensors that can be used to detect mobility in the city. Therefore, monitoring this data should be able to detect most of the important events in the city. [1]

4.2 DATASET AND FEATURES

Dataset is taken from the UCI Repository website for reference purpose, the link is given below [2]. http://mlr.cs.umass.edu/ml/datasets/Phishing+Websites

This dataset consists of totally 11055 websites 30 unique features represent each.

Out of this 11055. 60% of dataset for Training the classifier (6633) [[40% of dataset for Testing the classifier (4422)

Last column of the dataset is the Label of the instance i.e. “+1” or “-1” to say whether the instance dictating about the website is a phishing or non-phishing website

We will look at data from hour.csv, this data covers a 2 year period from 01/7/2021 to 31/7/2022

The resulting dataset that we will use contains 17389 instances and 17.

Attributes are of type integer and real.

The attributes are:

a. Moment: record index.

b. Dteday: current date.

c. Season: containing the integers 1 to 4 (1: spring, 2: summer, 3: autumn, 4: winter).

d. yr:year (0:2021, 1:2022).

e. mnth: month (1 to 12).

f. Hr: hour (0 to 23).

g. holiday: the weather is holiday or not (containing boolean expressions in

1 and 0 indicating whether the day of the observation is a public holiday or not)

h. Weekend: day of the week coded from (1 to 7).

i. working day: if the day is neither the weekend nor the public holidays, the value 1, otherwise, 0.

j. weathersit: containing the integers 1 to 4 representing four different lists of weather conditions:

i. Clear, few clouds.

ii. Broken Clouds, Fog.

iii. Light snow, light rain + thunderstorm.

iv. Heavy rain + ice pallets + fog, snow + fog

a) Time: normalized temperature in Celsius

b) Attemp: containing the values of the sensation temperature at the given moment, normalized in degrees Celsius.

c) Um: normalized humidity containing the values of the relative humidity level at the given moment.

d) Wind speed: containing the values of the normalized wind speed.

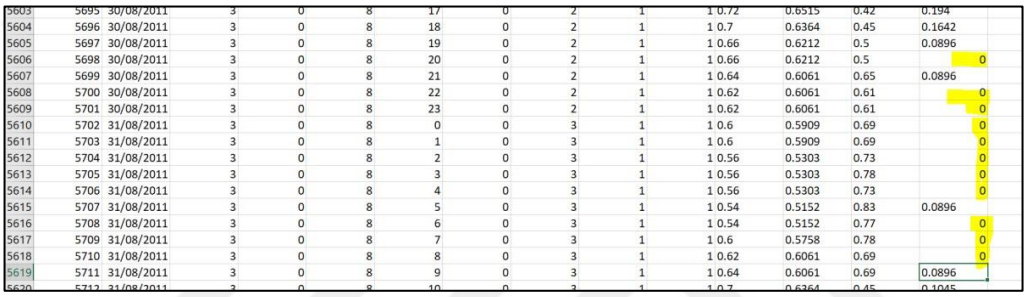
e) Casual: number of occasional users (not registered).

f) **Registered**: number of registered users.

g) **Cnt**: containing total number of bicycles rented, by occasional or registered users.

4.2.1 Data Cleaning

After analyzing the database we did a data cleaning, we deleted the variables:
“atempo”because it is almost repetitive and we keep the “temp” variable.
"casual"and 'registered' from the dataset as they total 'cnt'.
To better represent their categorical nature.
After cleaning and examining the DB we found that there are values = 0 in the sales speed variable of (hour.csv) so we directly removed the tuples containing these missing values to have an analysis accurate (we have a percentage of Noise fucked).



5603	5695	30/08/2011	3	0	8	17	0	2	1	1	0.72	0.6515	0.42	0.194
5604	5696	30/08/2011	3	0	8	18	0	2	1	1	0.7	0.6364	0.45	0.1642
5605	5697	30/08/2011	3	0	8	19	0	2	1	1	0.66	0.6212	0.5	0.0896
5606	5698	30/08/2011	3	0	8	20	0	2	1	1	0.66	0.6212	0.5	0
5607	5699	30/08/2011	3	0	8	21	0	2	1	1	0.64	0.6061	0.65	0.0896
5608	5700	30/08/2011	3	0	8	22	0	2	1	1	0.62	0.6061	0.61	0
5609	5701	30/08/2011	3	0	8	23	0	2	1	1	0.62	0.6061	0.61	0
5610	5702	31/08/2011	3	0	8	0	0	3	1	1	0.6	0.5909	0.69	0
5611	5703	31/08/2011	3	0	8	1	0	3	1	1	0.6	0.5909	0.69	0
5612	5704	31/08/2011	3	0	8	2	0	3	1	1	0.56	0.5303	0.73	0
5613	5705	31/08/2011	3	0	8	3	0	3	1	1	0.56	0.5303	0.78	0
5614	5706	31/08/2011	3	0	8	4	0	3	1	1	0.56	0.5303	0.73	0
5615	5707	31/08/2011	3	0	8	5	0	3	1	1	0.54	0.5152	0.83	0.0896
5616	5708	31/08/2011	3	0	8	6	0	3	1	1	0.54	0.5152	0.77	0
5617	5709	31/08/2011	3	0	8	7	0	3	1	1	0.6	0.5758	0.78	0
5618	5710	31/08/2011	3	0	8	8	0	3	1	1	0.62	0.6061	0.69	0
5619	5711	31/08/2011	3	0	8	9	0	3	1	1	0.64	0.6061	0.69	0.0896
5620	5712	31/08/2011	3	0	8	10	0	3	1	1	0.7	0.6364	0.45	0.1045

Figure 4.1: Data from day.csv

The result of data cleaning is a dataset with 15199 observations and 12 variables, the following figure shows the data structure after cleaning.

4.2.2 Phishing Data Set website

Phishing is a backdoor approach used by cyber crooks targeting users, not computers, to obtain sensitive or confidential information such as passwords, credit card numbers, social security numbers, or bank account numbers. [2]
There are different phishing techniques such as phishing by email, messages, SMS and website; however we will limit ourselves to websites in order to carry out our project.
Phishing steals identities and destroys lives. It affects everyone, from a bank manager to someone who has never heard of Internet scams. What's worse is that even though it's now over 10 years old, many people don't know about it.
Now let's look at a few aspects to show us the impact or damage caused by phishing on the Internet.

Although phishing can have harmful consequences in various ways, we will look in particular at the aspects of financial loss, reputational damage and their consequences in terms of denial of service.

a. Companies must absolutely protect themselves from all forms of threats, especially for sensitive data.

Indeed attempts to click on phishing URLs have increased by 85% since 2021 which represents a considerable increase, so statistics show that more than 300 Million phishing URLs have been detected this year alone.

So the way businesses protect themselves against phishing attempts needs to change as well.

b. Phishing messages appear to come from legitimate organizations such as government or your bank; however, these are actually clever scams. The messages politely ask for account information to be updated, validated, or confirmed, frequently suggesting that something went wrong. You are then redirected to a fake site where you are tricked into entering account information. This can result in identity theft.
c. The Cost of a Damaged Brand Reputation While the financial damages can be recovered in no time, it is the damage to a brand's reputation that takes years to trace back to where it originated. If something goes wrong, customers are less likely to do business with you in the future.

4.3 IMPLEMENTATION

The main backbone for the creation of these algorithms is MATLAB

MATLAB is a computational software used to work with mathematically and statically registered data. It uses computational workspace to find the result of the command executed in the command prompt at each step. It helps us use the inbuilt library as well as use scratch for new function creation, there are even inbuilt function for Naïve Bayes, Decision tree, SVM, Etc. MATLAB version 2010 till 2016 can be used for this classification. Other than MATLAB software’s like SVM library, WEKA machine learning software’s can also be used. Preference for MATLAB is that it does fast computation even with higher order of matrix compared to other Machine Learning software’s.

The essential objectives of research and detection of phishing websites are:

a. Protect companies against identity theft.

a. Protect users from scamming counterfeit sites.
b. Reduce users' precautions and give them a feeling of security.

To achieve these objectives and within the framework of this mini-project we will study phishing detection data from websites, and determine the factor that determines whether a website is a source of danger (phishing) or a secure site.

Our exploration and data analysis will be performed in MATLAB as requested.

We are going to examine the data from the DB which has the following fields:

The resulting dataset that we will use contains 1353 instances and 10 variables.

Attributes have values (-1, 0 or 1).

We first copied our DB into an Excel file for better visualization and data manipulation as shown in Figure 2:

	SFH	popUpWidnow	SSLfinal_State	Request_URL	URL_of_Anchor	eb_traffic	URL_Length	age_of_domain	having_IP_Address	Result
2	1	-1	1	-1	-1	1	1	1	0	0
3	-1	-1	-1	-1	-1	0	1	1	1	1
4	1	-1	0	0	-1	0	-1	1	0	1
5	1	0	1	-1	-1	0	1	1	0	0
6	-1	-1	1	-1	0	0	-1	1	0	1
7	-1	-1	1	-1	-1	1	0	-1	0	1
8	1	-1	0	1	-1	0	0	1	0	-1
9	1	0	1	1	0	0	0	1	1	-1
10	-1	-1	0	-1	-1	-1	-1	1	0	0
11	-1	0	-1	-1	1	1	0	-1	0	1
12	-1	-1	0	-1	-1	1	-1	-1	0	1

Figure 4.2: Data from phishing.csv

The attributes are:

a. SFH (Server Form Handler): if the content of a string is empty or "about: blank “suspect 0, otherwise if the Domain SFH name is the same as the domain name then 1 legitimate, otherwise -1 phishing.

b. Pop-up Window: if the popup asks for personal information then phishing -1, otherwise legit 1.

c. SSL state: if a site uses an SSL certificate it means legitimate 1, otherwise -1, but in our current time with the disturbing development the whole site can be suspect 0.

d. Request URL: if the external objects contained in the website (image, video, sound, etc.) have the same domain name, this means 1 (legitimate) otherwise if the domain name of the objects and websites are undone then -1 (phishing).

e. Anchor URL: This is the element defined by the tag <a> it takes the value 1 if %de URL ANCHOR <31%□legit. 0 if 31 %< =%of URL ANCHOR<=67%□suspicious -1 otherwise.

f. Website traffic: if the website is classified in the Alexa database it means legitimate 1, otherwise if it is classified among the first 100000 which are not in the Alexa database then it is suspicious 0, otherwise it is necessarily phishing 1.

g. URLLength: if the length of the URL is less than or equal to 54 characters it means 1 legitimate, and if it is between 54 and it's around 0 suspicious, otherwise -1 phishing.

h. Domain Age: If the Age of a domain name is more than a year it is surely legitimate 1, otherwise if its age is between 1 year and 6 months□0 "Suspect", -1 otherwise.

i. IP: If an IP address is found (even sometimes in hexadecimal) then -1 phishing otherwise 1 legitimate.

j. Results: the result of classification of websites 1□legit, 0□suspicious or -1□phishing.[3]

4.4 ALGORITHMS USED

4.4.1 Logistic Regression

Logistic regression is a supervised algorithm classification, popular in Machine Learnin widely used because it allows to model binary variables or sums of binary variables. [2]

a. Advantages

The outputs have an interesting probabilistic interpretation and the algorithm can be regularized to avoid over-adjustments. Logistic models can be easily updated with new data using stochastic gradient descent.

b. Inconveniences

Logistic regression tends to underperform in the presence of multiple or nonlinear decision limits. They are not flexible enough to naturally capture more complex relationships. [3]

Our dataset is composed of 561 rows and 10 columns, we used 70% of this first one for learning which is 392 rows and the remaining 30% will be used for testing.

We have initialized a vector labels which contains the unique values of our dataset and which represent the classification classes 0, -1, 1.

Then we devised our dataset as we mentioned before in 2 parts 70% for learning by storing input values in Xtrain and output values in Ytrain, 30% for testing by storing input values in Xtest and the output values in Ytest.

After that we normalized the data using the normalize function.

We have initialized the values alpha, iter, lambda which respectively represent the convergence pat used to calculate the downward gradient, the number of iterations of the same function, and the regularization parameter.

Then we called the prediction function which has as input Xtrain, Ytrain, Xtest, labels, alpha, iter and lambda that we have already initialized and gives us in return the best theta value, the predictions, and the cost.

After initializing the size of the training part, we created an all_theta matrix which will contain the theta values returned by the gradient Descent for each class, so it contains a number of rows equal to the number of labels.

We also initialized the prediction vector which will contain the predictions as well as other necessary initializations, then we calculated the theta for each class c by setting it to 1 and the other classes to 0 to finally save them in the all theta matrix.

And for that we used the function gradient Descent in figure (1):

```
function [theta , vect_cost] = gradientDescent(X, y, theta, alpha, iteration,lambda)
  m = length(y);
  vect_cost = zeros(iteration, 1);
  n = length(theta);
  temp = theta;
  for it = 1 : iteration
    erreur = (hypothese(X * theta) - y);
    for j=1 : n
        temp(j,1) = sum(erreur.*X(:,j));
    end
    theta = theta - (alpha/m) * temp + (lambda/m) * temp;
    vect_cost(it,1) = cost(X,y,theta,lambda);
  end
end
```

Figure 4.3: code for gradient Descent

In order to avoid 'overfitting' and 'underfitting' we used the regularization parameter 'lambda' which affected the cost and gradient Descent calculations.

Returning to the prediction class, we calculated the probabilities of each class and for this we used the hypothesis function which contains a simple mathematical calculation already seen in class.

```
function [h]=hypothese(e)
  % Calcule de l'hypothese
  h = (1 ./ (1 + exp(-e)));
end
```

Figure 4.4: Calculating the LR function

We used these probabilities to find the prediction which is the index of the line with the highest hypothesis, this prediction allowed us to calculate the accuracy of our learning.

Although we normalized the data and modified our database to reduce noisy data, and we varied our lambda and the number of iterations, we had an accuracy of

53.25%

Finally, we display the graph of the cost function which gives the following result (figure 3):

4.4.2 Neural Networks

The neural network itself is not an algorithm, but rather a framework for many machine learning algorithms to work together and process complex data inputs.

The artificial neural network consists of a set of processing elements that are highly interconnected and transform a set of inputs to a set of desired outputs. The result of the transformation is determined by the characteristics of the elements and the weights associated with the interconnections among them. A neural network performs an analysis of information and provides a probability estimate that it matches the data it has been trained to recognize.

In order to implement our solution, we made a small design following the notions seen in

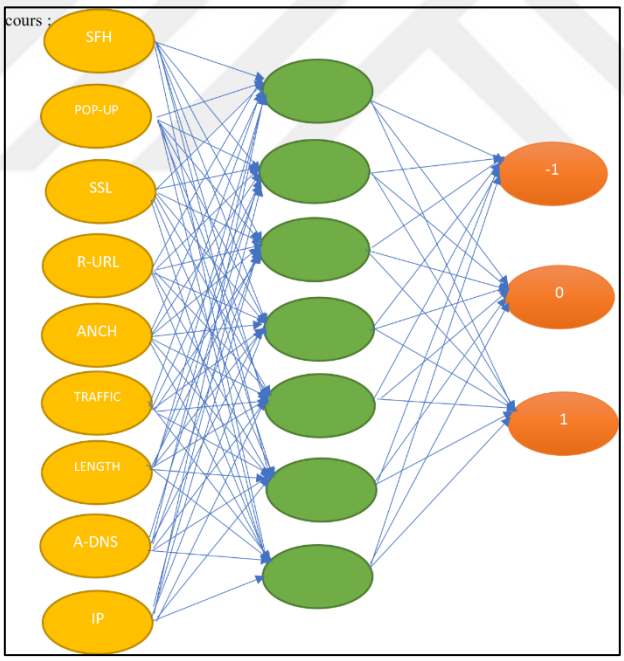


Figure 4.5: Ann Input Layer Hidden-Layer Output

First we imported the data from the already improved database, we separated the inputs with a selection of columns and we did the same with the results column Y, then we defined the number of inputs (9 in our case), and the number of hidden layer by 7 (following the 75% rule), and finally we set the number of class to 3 (-1 0 and 1).

Thereafter we initialized theta 1 and theta 2 with the function (randWeight_Team02) which takes as input the Layer input and output and as output it initializes the weights of layer input in a random way.

After we have transformed the theta matrices into a flat vector.

The function cost will first reshape the variable params in the parameters of theta1 and 2, Feedforward the neural network and return the cost in the variable J, do the regularization and implement it with gradient and cost.

The cost function also makes the call to the sigmoid function seen already in progress.

We calculated the cost for 50 iterations, and we found that the cost is decreasing which is good.

We also used the Fmincing function which performs a gradient descent.

Finally we retrieved the values of theta 1 and theta 2 from the nn_params parameter to use them in the precision calculation with the predict function and we obtained a precision equal to 46.702317

4.4.3 Support Vector Machine (SVM)

Support Vector Machines (SVMs) use a mechanism called kernels, which essentially calculate the distance between two observations. The SVM algorithm then finds a decision boundary that maximizes the distance between the closest members of the separate classes.

For example, an SVM with a linear kernel is similar to logistic regression. Therefore, in practice, the advantage of SVMs usually comes from using nonlinear kernels to model nonlinear decision limits.

a. **Advantages**

SVMs can model nonlinear decision limits, and there are many kernels to choose from. They are also quite robust against overfitting, especially in large spaces.

b. **Disadvantages**

However, SVMs require a lot of memory, are more difficult to tune due to the importance of choosing the right kernel, and do not scale well to larger data sets. Currently in the industry, random forests are generally preferred over SVMs.

c. **Implementation**

In a first step we did the preprocessing of data by importing them by putting the incoming data in the variable X and the outgoing ones in Y, then we broke down our "dataset" into two parts, 70% for l learning from our data and 30% for testing.

Then we called the functionfitcecoc" and store the result of this function which is of type "ECOC" in the model variable and which is formed from the predicate data set of X and the Y. After the model recording, we called the predict function which will predict the data set which is in model and the one in X_test to save them in y_prediction.

Finally, we displayed the data precision by calculating the similarity between the old values of Y_test and the new predicate values obtained in the previous line.

After executing the code, we obtained an accuracy of 73.39% as shown in the following figure (4.6):

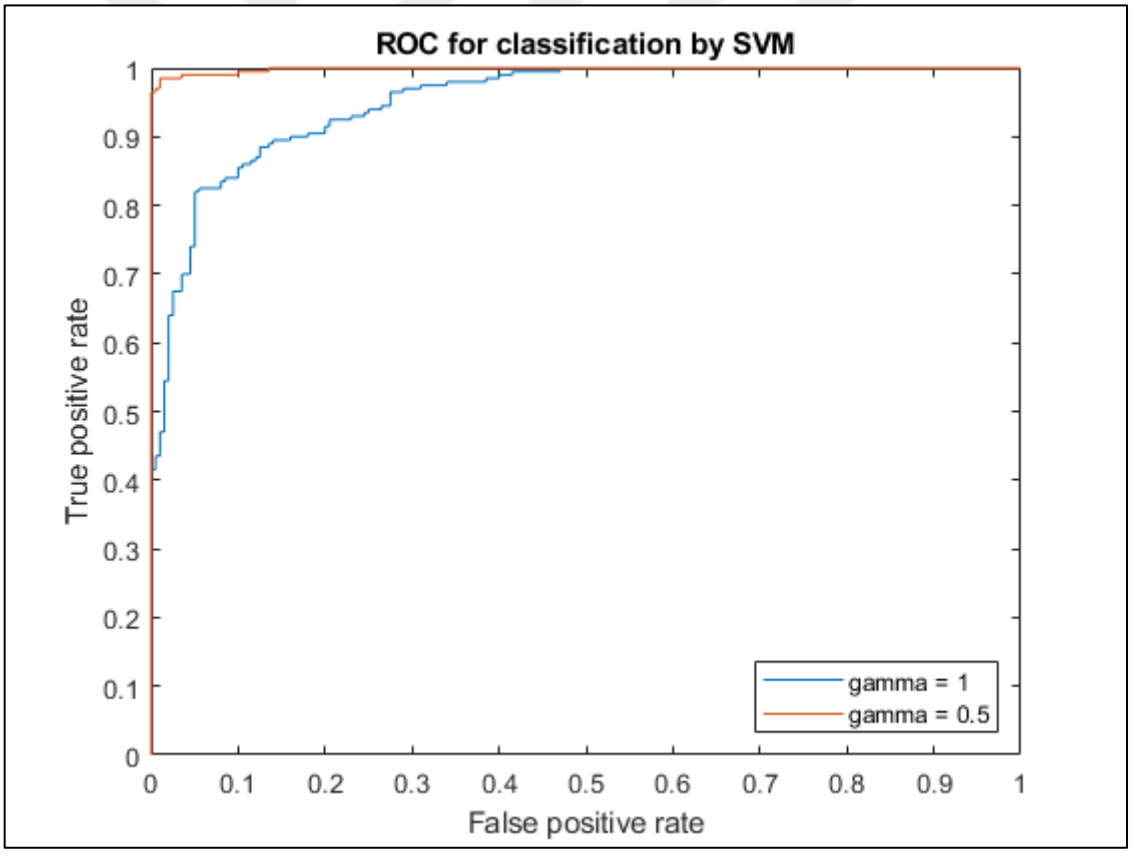


Figure 4.6: SVM Precision

4.4.4 Decision Trees DT

Decision trees are a type of supervised machine learning, the tree can be explained by two or more entities, namely decision nodes and leaves. The leaves are the decisions or the final results in our case they are the three classes (legitimate [1], suspicious [0] and phishing [0]), and the decision nodes are where the data is stored. [6]

A. Advantage

i. Simple is a model that is simple to explain a situation using boolean logic

ii. We can use it even without normalizing our dataset or modifying it Validation is done using a static test, to see the reliability of a model

B. Disadvantages

i. We can have a very complex tree (relationship with attributes)

ii. Not all can be expressed using decision trees (XOR)

C. Implementation

Like all the other models we start importing our dataset (just 70% for learning), our X contains the 9 attributes and the Y contains the result classes (1,0,-1)

We follow we use the predefined function fitctree which takes as input the attributes X and Y the results of the latter to build and the function view display our tree The function fitctree:

```
tree = fitctree(x,y);
```

The view function:

```
view (tree,'mode','graph');
```

The result is shown in figure (4.7).

In the diagram we notice that the leaves are our class and the knots are the attributes

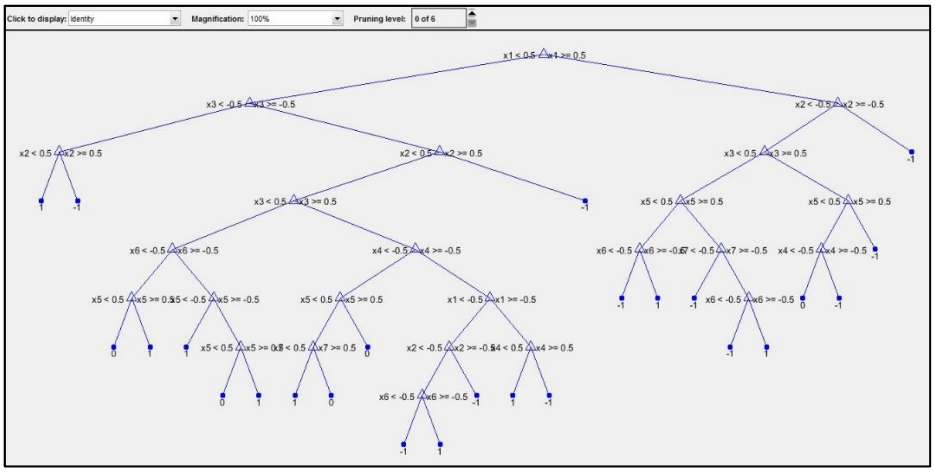


Figure 4.7: DT workflow in MATLAB

Is to test our tree we can use static tests using the predict function which takes our tree as input, and a tuple of our choice (from the last 30% dedicated for the tests)

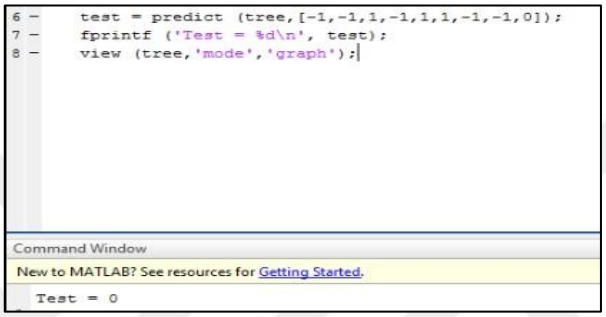
```
test = predict (tree,[1,1,-1,-1,1,-1,0,1,0]);
```

For the prediction of our learning rate of our decision tree model, we did several static tests and we rarely came across confluences between classes, for example the following tests

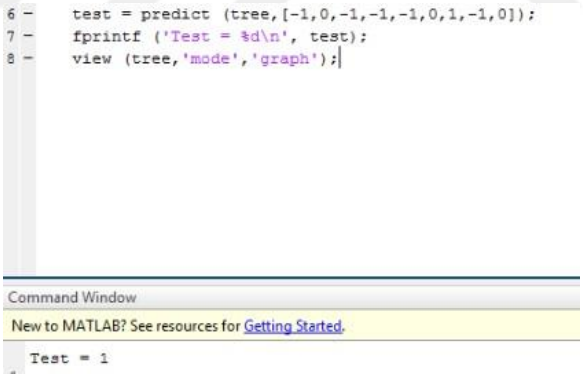
Test 1: the X [1 1 -1 -1 1 -1 0 1 0] and the result according to our dataset and -1

Test 2: the X [-1 -1 1 -1 1 1 -1 -1 0] and the result according to our dataset and 0

The result according to our model is (figure 4.8):



Test 3: the X [-1 0 -1 -1 -1 0 1 -1 0] and the result according to our dataset is 1 the result according to our model is in figure (9)



4.5 RESULTS DISCUSSION

To find whether the given website is phishing or non-phishing we need to build a classifier based on certain predicted data and use the classifier for validation purpose to confirm whether the build classifier is efficient enough to predict the future driven website. This classifier is built using three algorithms Naïve Bayes, Decision tree and Support vector machines. Since the Dataset has only few instances therefore we have high bias and low variance, So Naïve Bayes can be used including with this to check the highest accuracy rate addition of two more algorithms are used to find which algorithm has the best accuracy.

Secondarily to increase accuracy of the classifier we reduce the number of features i.e. feature selection technique to identify which set of features are reasonable enough to achieve best accuracy for the classifier. So, that the classifier with highest accuracy can predict the future developed or newly added website correctly. Thus, the classifier with highest accuracy acts the boundary for classifying the website this barrier contains the requirements of non-phishing website thus acts as a tester.

Why Different algorithms have different classification accuracy?

Naïve Bayes: Naive Baye technique uses Bayes theorem as the base for building classifier model. Bayes theorem states that the prediction of new event can be obtained if the relevant or related feature knowledge is obtained and if the probability of the event seems to be greater than 0.5 then it seeks to be the correct prediction. Mathematically Bayes theorem is represented by,

Naïve Bayes is the fastest technique to predict data but doesn't work for large amount of training data. Addition to the above-mentioned limitation Naïve Bayes also assumes predicting features to be independent but in real time applications it is difficult to get those kinds of feature set.

Decision Tree: Decision tree builds classifier as tree structure the entropy formula determines every node and leaf of the tree

The feature with highest entropy is the root node. The feature with second highest entropy is considered as the child of the root node by considering the entropies of the other features the tree is formed. To shorten the tree length pruning is done i.e. removing the features that gives less contribution to the tree classifier. Entropy of a feature defines how efficient is the feature to classify the instance whether it is true or false in our case whether the website is phishing or non-phishing. The main con of this decision tree technique is that each time a new example is created the decision tree should be reformed again so that it might overfit.

Support Vector Machines: This technique builds a hyperplane or boundary based on discriminant function. This hyperplane categorizes the data into two group's one on one side of the boundary and the other on second side, the model represents the data in space as points. The boundary can be a linear boundary or non-linear boundary which is later created as linear boundary in difference space using kernel trick. Also, SVM supports unsupervised data's also by support vector clustering which clusters similar data into same groups.

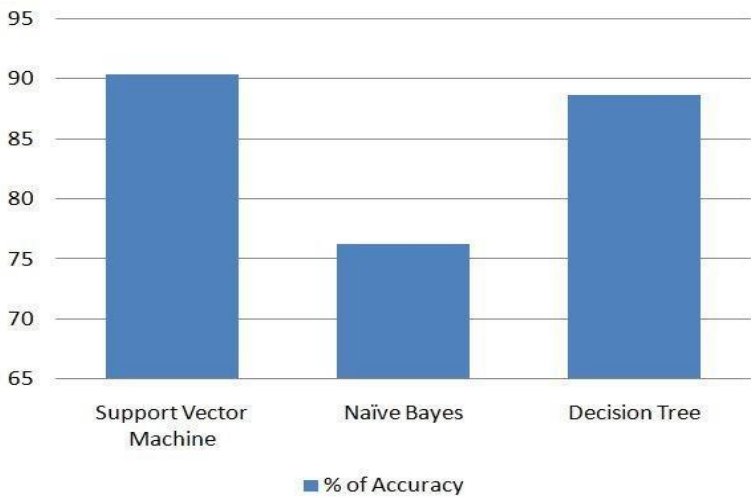


Figure 4.8: SVM, NB, and DT accuracy compared

There are two major advantages of SVM they are:

a. High accuracy

b. Guarantees theoretically regarding under fitting and overfitting

Feature Selection: Since data set contains 31 features considering all the features for building a classifier might reduce the accuracy as well as time computation. Therefore, reducing the number of features to build a classifier increases both accuracy and time for building classifier. In this project forward feature selection technique is used i.e. finding a single feature with maximum accuracy and adding features to that single feature in order to increase the accuracy of the classifier, Thus at a single point of time the feature set starts reducing the accuracy that point defines the actual number of features required to build a classifier and what features required to build it to achieve maximum accuracy.

Table 4.1: Features according to the weights in the dataset

Total number of features	**Feature Number**
having_IP_Address	1
URL_Length	2
Shortining_Service	3
having_At_Symbol	4
double_slash_redirecting	5
Prefix_Suffix	6
having_Sub_Domain	7
SSLfinal_State	8
Domain_registeration_length	9
Favicon	10
port	11
HTTPS_token	12
Request_URL	13
URL_of_Anchor	14
Links_in_tags	15
SFH	16
Submitting_to_email	17
Abnormal_URL	18
Redirect	19
on_mouseover	20
RightClick	21
popUpWidnow	22
Iframe	23
age_of_domain	24
DNSRecord	25
web_traffic	26
Page_Rank	27
Google_Index	28
Links_pointing_to_page	29
Statistical_chapter	30

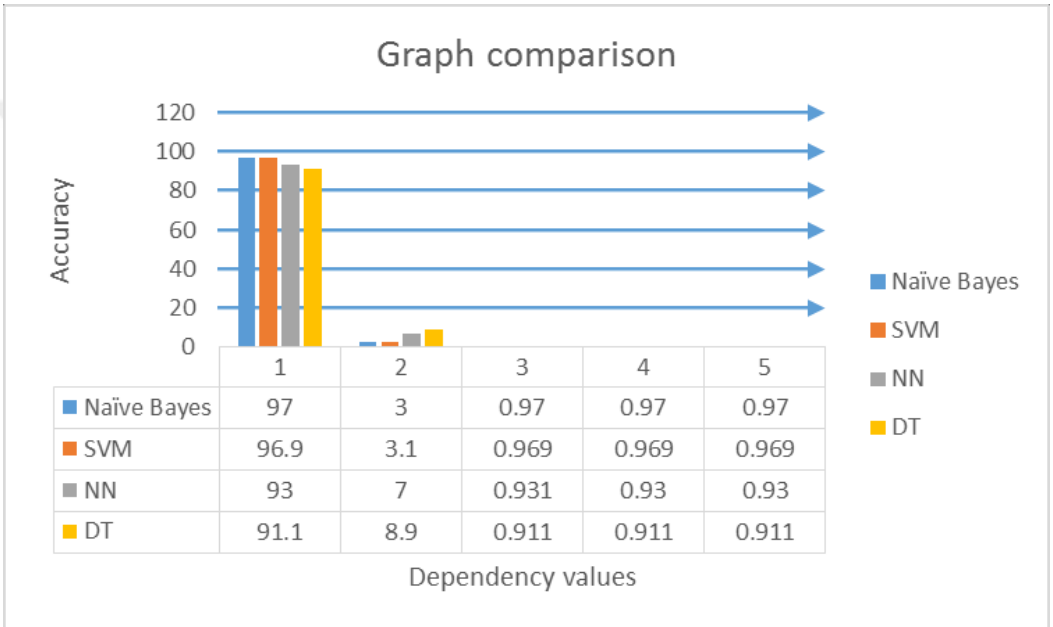
Out of these 30 features the required set of features for achieving maximum accuracy is

Feature set= {8, 14, 6, 15, 2, 7, 9, 16, 13, 15, 28, 29, 18}

Target accuracy with all features=91.45% **(SVM)**

Feature set Accuracy= 94.2108% (**ANN-Naïve Bayes)**

Target accuracy with all features=89.1904% (**Decision Tree)**



	1	2	3	4	5
Naïve Bayes	97	3	0.97	0.97	0.97
SVM	96.9	3.1	0.969	0.969	0.969
NN	93	7	0.931	0.93	0.93
DT	91.1	8.9	0.911	0.911	0.911

Figure 4.9: SVM, NB, DT, and NN error rate comparison

5. CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

Phishing attacks have significantly decreased as a result of significant contributions made by the academic community. The goal of this research is to investigate three common types of fraudulent activity that can be found online and to examine potential preventative strategies that are based on machine learning and neural networks. In addition to that, quite a few fundamental ideas concerning NI techniques, ML, ANN, and email filtering are presented. In addition, some of the basic concerns that are associated with the detection of e-fraud are discussed, along with some methods that can be implemented to increase the efficiency of the detection systems that have been researched. In addition, two improved techniques for recognizing phishing emails have been provided. The RF methodology forms the foundation of the first approach, while the ANN and SVM modelling approaches are brought together in the second approach (termed ANN SVM). Through the application of the second method, the FFA and SVM models are combined into a single comprehensive model. First, an ANN is utilized to determine the most effective combination of hyper-parameters, and then a support vector machine is used to carry out the classification. The development of a hyper-parameter selection technique is the primary focus of this body of work, with the end goal of enhancing the ANN-SVM's capacity for accuracy and functionality. The length of the C and value sequence that is produced by this method grows with each repetition of the method. In order to determine which of the two approaches offers the greater overall value, a series of comparisons are made using datasets of increasing sizes. Both the RF-based classifier and the SVM-based classifier based on fireflies are used in the experiments that are carried out. The results of the investigations offered a great number of reasons to have positive expectations. The RF-based classifier had an accuracy of classification of 99.70%, whereas the ANN SVM had an accuracy of classification of 99.99%. In addition, the DT-based classifier produced a false positive rate of 0.06% and a false negative rate of 2.50%, whereas the ANN SVM produced a false positive rate of 0.01% and a rate of 0% for false negatives.

It is impossible to overstate the significance of having a comprehensive electronic fraud detection system that is also safe, quick, and accurate in light of the enormous monetary losses that have been incurred by a number of companies and individuals as a direct result

of fraudulent activity that has taken place online. When doing research in the subject of e-fraud detection, researchers frequently come up against a variety of challenges, the most notable of which is the rapid change of fraud patterns. Phishing, unsolicited email, and unauthorized access to computer systems and networks are the three most common types of fraudulent activity that can occur online. The most important purpose of this research was to figure out how to recognize phishing tactics. Phishing continues to be a huge threat to the expansion and stability of economies all over the world, despite the fact that it has already taken advantage of a substantial number of individuals and businesses. It is a thriving industry that will continue to expand over time because of the exponential rise of technology, and it is an industry that will continue to increase over time. It has proven to be difficult to keep trustworthy blacklists up to date because of the un-centralized nature of phishing efforts and the ongoing growth of new phishing sites. As a consequence of this, there is an immediate requirement for a fraud detection system that is flexible enough to accommodate frauds that are always evolving. DT, ANN, and SVM were utilized in the construction of two separate classifiers for the purpose of recognizing phishing emails in this research. Experiments that were carried out on these classifiers show that they have a powerful filtering capability and are able to correctly identify phishing emails. In addition, the classifiers can be altered to take into account newly discovered phishing methods as they come to light. Both of the solutions that have been recommended for phishing emails can easily be implemented into existing email filters. This will unquestionably reduce the number of phishing attempts and make the internet environment more secure for companies and individuals who engage in a variety of online transactions.

5.2 FUTURE WORK

To protect customers from increasingly sophisticated phishing scams, the detection systems that are now available will need to undergo significant development. According to the findings of the research, SVM is painfully slow when it comes to categorizing, particularly when it comes to dealing with big amounts of data. This is because, in order to categorize a single data sample, the kernel function is normally evaluated for each of the support vectors that have been defined. This is the reason why this is the case. Because of the need for quick processing, this cannot under any circumstances be considered a viable option for fraud detection systems. An improved support vector machine (SVM) model is going to be constructed at some point in the not-too-distant future. Finding strategies to cut down on the

total amount of time the model is required to spend being trained will be a primary focus. The integration of a greater variety of NI and ML approaches is one of the goals that we have set for ourselves moving forward. It is possible that the performance of hybrid systems will be better than that of traditional ones.

REFERENCES

[1] Y. Zhang, Y. Xiao, K. Ghaboosi, J. Zhang, and H. Deng, ‘‘A survey of cyber crimes,’’ *Secur. Commun. Netw.*, vol. 5, no. 4, pp. 422–437, 2012.

[2] APWG Developers. (2021). *Phishing Activity Trends Report*. [Online]. Available: https://apwg.org/trendsreports/

[3] M. Lei, Y. Xiao, S. V. Vrbsky, and C.-C. Li, ‘‘Virtual password using random linear functions for on-line services, ATM machines, and pervasive computing,’’ *Comput. Commun.*, vol. 31, no. 18, pp. 4367–4375, Dec. 2008.

[4] P. Burda, L. Allodi, and N. Zannone, ‘‘Don’t forget the human: A crowdsourced approach to automate response and containment against spear phishing attacks,’’ in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS PW)*, Sep. 2020, pp. 471–476.

[5] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, ‘‘PhishNet: Predictive blacklisting to detect phishing attacks,’’ in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–5.

[6] W. Zhang, Y.-X. Ding, Y. Tang, and B. Zhao, ‘‘Malicious web page detection based on on-line learning algorithm,’’ in *Proc. Int. Conf. Mach. Learn. Cybern.*, vol. 4, Jul. 2011, pp. 1914–1919.

[7] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. González, ‘‘Classifying phishing URLs using recurrent neural networks,’’ in *Proc. APWG Symp. Electron. Crime Res. (eCrime)*, 2017, pp. 1–8.

[8] B. Cui, S. He, X. Yao, and P. Shi, ‘‘Malicious URL detection with feature extraction based on machine learning,’’ *Int. J. High Perform. Comput. Netw.*, vol. 12, no. 2, pp. 166–178, 2018.

[9] Y. Fang, C. Zhang, C. Huang, L. Liu, and Y. Yang, ‘‘Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism,’’ *IEEE Access*, vol. 7, pp. 56329–56340, 2019.

[10] J. Feng, L. Zou, O. Ye, and J. Han, ‘‘Web2Vec: Phishing webpage detection method based on multidimensional features driven by deep learning,’’ *IEEE Access*, vol. 8, pp. 221214–221224, 2020.

[11] H. Cheng, J. Liu, T. Xu, B. Ren, J. Mao, and W. Zhang, ‘‘Machine learning based low-rate DDoS attack detection for SDN enabled IoT networks,’’ *Int. J. Sens. Netw.*, vol. 34, no. 1, pp. 56–69, 2020.

[12] S. Christin, É. Hervet, and N. Lecomte, ‘‘Applications for deep learning in ecology,’’ *Methods Ecol. Evol.*, vol. 10, no. 10, pp. 1632–1644, Oct. 2019.

[13] A. Aggarwal, A. Rajadesingan, and P. Kumaraguru, ‘‘PhishAri: Automatic realtime phishing detection on Twitter,’’ in *Proc. eCrime Res. Summit*, Oct. 2012, pp. 1–12.

[14] H. Ma, Y. Zuo, and T. Li, ‘‘Vessel navigation behavior analysis and multiple-trajectory prediction model based on AIS data,’’ *J. Adv. Transp.*, vol. 2022, pp. 1–10, Jan. 2022.

[15] J. Fang, B. Li, and M. Gao, ‘‘Collaborative filtering recommendation algorithmbasedondeepneuralnetworkfusion,’’*Int.J.Sens.Netw.*,vol.34, no. 2, pp. 71–80, 2020.

[16] E. S. Gualberto, R. T. De Sousa, T. P. De Brito Vieira, J. P. C. L. Da Costa, and C. G. Duque, ‘‘The answer is in the text: Multi-stage methods for phishing detection based on feature engineering,’’ *IEEE Access*, vol. 8, pp. 223529–223547, 2020.

[17] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, ‘‘Short-term residential load forecasting based on LSTM recurrent neural network,’’ *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.

[18] E. Zhu, Y. Chen, C. Ye, X. Li, and F. Liu, ‘‘OFS-NN: An effective phishing websites detection model based on optimal feature selection and neural network,’’ *IEEE Access*, vol. 7, pp. 73271–73284, 2019.

[19] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, ‘‘Phishing email detection using natural language processing techniques: A literature survey,’’ *Proc. Comput. Sci.*, vol. 189, pp. 19–28, Jan. 2021.

[20] M. Korkmaz, O. K. Sahingoz, and B. Diri, ‘‘Feature selections for the classification of webpages to detect phishing attacks: A survey,’’ in *Proc. Int. Congr. Hum.-Comput. Interact., Optim. Robotic Appl. (HORA)*, Jun. 2020, pp. 1–9.

[21] C. Singh and Meenu, ‘‘Phishing website detection based on machine learning: A survey,’’ in *Proc. 6th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Coimbatore, India, 2020, pp. 398–404. [Online]. Available: https://ieeexplore.ieee.org/document/9074400/authors#authors, doi: 10.1109/ICACCS48705.2020.9074400.

[22] L. Tang and Q. H. Mahmoud, ‘‘A survey of machine learning-based solutions for phishing website detection,’’ *Mach. Learn. Knowl. Extraction*, vol. 3, no. 3, pp. 672–694, Aug. 2021.

[23] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, ‘‘A comprehensive survey of AI-enabled phishing attacks detection techniques,’’ *Telecommun. Syst.*, vol. 76, pp. 139–154, Oct. 2020.

[24] M. Vijayalakshmi, S. M. Shalinie, M. H. Yang, and M. R. Meenakshi, ‘‘Web phishing detection techniques: A survey on the state-of-the-art, taxonomy and future directions,’’ *IET Netw.*, vol. 9, no. 5, pp. 235–246, Sep. 2020.

[25] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, ‘‘Deep learning for phishing detection: Taxonomy, current challenges and future directions,’’ *IEEE Access*, vol. 10, pp. 36429–36463, 2022.

[26] A. Odeh, I. Keshta, and E. Abdelfattah, ‘‘Machine LearningTechniquesfor detection of website phishing: A review for promises and challenges,’’ in *Proc. IEEE 11th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2021, pp. 0813–0818.

[27] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, ‘‘A systematic literature review on phishing email detection using natural language processing techniques,’’ *IEEE Access*, vol. 10, pp. 65703–65727, 2022.

[28] T. Mahjabin, Y. Xiao, T. Li, and C. L. P. Chen, ‘‘Load distributed and benign-bot mitigation methods for IoT DNS flood attacks,’’ *IEEE Internet Things J.*, vol. 7, no. 2, pp. 986–1000, Feb. 2020.

[29] W. Yang, W. Zuo, and B. Cui, ‘‘Detecting malicious URLs via a keywordbased convolutional gated-recurrent-unit neural network,’’ *IEEE Access*, vol. 7, pp. 29891–29900, 2019.

[30] M. Somesha, A. R. Pais, R. S. Rao, and V. S. Rathour, ‘‘Efficient deep learning techniques for the detection of phishing websites,’’ *Sadhan¯a¯*, vol. 45, no. 1, pp. 1–18, Dec. 2020.

[31] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J.-P. Niyigena, ‘‘An effective phishing detection model based on character level convolutional neural network from URL,’’ *Electronics*, vol. 9, no. 9, p. 1514, Sep. 2020.

[32] Google Developers. (2020). *Google Site*. [Online]. Available: https://sites.google.com/

[33] Phishtank Developer. (2020). *Phishtank Dataset*. [Online]. Available: https://phishtank.org/developerinfo.phpl

[34] Alexa Developer. (2020).*Alexa Dataset*. [Online]. Available: https://www.alexa.com/topsites

[35] JoeWein Developer. (2020). *JoeWein Dataset*. [Online]. Available: https://joewein.net/bl-log/bl-log.html

[36] Common Crawl Developer. (2020). *Common Crawl Dataset*. [Online]. Available: https://commoncrawl.org/2021/10/september-2021-crawlarchive-now-available/

[37] OpenPhish Developer. (2020). *OpenPhish Dataset*. [Online]. Available: https://openphish.com

[38] A. Draghetti. (2020). *Phishing Army: The blocklist to Filter Phishing!* [Online]. Available: https://www.phishing.army/

[39] S. Ghodke. (Jan. 2018). *Alexa Top 1 Million Sites*. [Online]. Available: https://www.kaggle.com/datasets/cheedcheed/top1m

[40] Accessed: Jul. 10, 2021. [Online]. Available: https://archive.ics.uci.edu/ml//index.php

[41] G. Vrbančič, I. Fister, and V. Podgorelec, ‘‘Datasets for phishing websites detection,’’ *Data Brief*, vol. 33, Dec. 2020, Art. no. 106438. [Online]. Available: https://www.sciencedirect.com/science/article/pii/ S2352340920313202

[42] G. Sood. (Feb. 2021). *Parsed DMOZ Data*. [Online]. Available: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi%3A10.7910%2FDVN%2FOMV93V

[43] *Webscope | Yahoo Labs*. Accessed: Aug. 10, 2021. [Online]. Available: https://webscope.sandbox.yahoo.com/

[44] *Yandex.xml*. [Online]. Available: https://yandex.com/dev/xml/

[45] Accessed: Aug. 10, 2021. [Online]. Available: https://www.medien.ifi.lmu.de/team/max.maurer/files/phishload/

[46] D. Nayak and H. Perros, ‘‘Automated real-time anomaly detection of temperaturesensorsthroughmachine-learning,’’*Int.J.Sens.Netw.*,vol.34, no. 3, pp. 137–152, 2020.

[47] M. Guo, C. Guo, C. Zhang, D. Zhang, and Z. Gao, ‘‘Fusion of ship perceptual information for electronic navigational chart and radar images based on deep learning,’’*J. Navigat.*, vol. 73, no. 1, pp. 192–211, Jan. 2020.

[48] N. Wu, X. Wang, B. Lin, and K. Zhang, ‘‘A CNN-based end-to-end learning framework toward intelligent communication systems,’’ *IEEE Access*, vol. 7, pp. 110197–110204, 2019.

[49] J. Atiga, M. Hamdi, R. Ejbali, and M. Zaied, ‘‘Recurrent neural network NARX for distributed fault detection in wireless sensor networks,’’ *Int. J. Sens. Netw.*, vol. 37, no. 2, pp. 100–111, 2021.

[50] F. Tajaddodianfar, J. W. Stokes, and A. Gururajan, ‘‘Texception: A character/word-level deep learning model for phishing URL detection,’’ in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2857–2861.

[51] H. Le, Q. Pham, D. Sahoo, and S. C. H. Hoi, ‘‘URLNet: Learning a URL representation with deep learning for malicious URL detection,’’ 2018, *arXiv:1802.03162*.

[52] J. Yuan, G. Chen, S. Tian, and X. Pei, ‘‘Malicious URL detection based on a parallel neural joint model,’’ *IEEE Access*, vol. 9, pp. 9464–9472, 2021.

[53] M. Sameen, K. Han, and S. O. Hwang, ‘‘PhishHaven—An efficient real-time AI phishing URLs detection system,’’ *IEEE Access*, vol. 8, pp. 83425–83443, 2020.

[54] Y. Huang, Q. Yang, J. Qin, and W. Wen, ‘‘Phishing URL detection via CNN and attention-based hierarchical RNN,’’ in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2019, pp. 112–119.

[55] S.-J. Bu and S.-B. Cho, ‘‘Deep character-level anomaly detection based on a convolutional autoencoder for zero-day phishing URL detection,’’ *Electronics*, vol. 10, no. 12, p. 1492, Jun. 2021.

[56] U. Kamath, J. Liu, and J. Whitaker, *Deep Learning for NLP and Speech Recognition*, vol. 84. Cham, Switzerland: Springer, 2019.

[57] N. Cao, W. Huo, T. Lin, and G. Wu, ‘‘Application of convolutional neural networks and image processing algorithms based on traffic video in vehicle taillight detection,’’ *Int. J. Sens. Netw.*, vol. 35, no. 3, pp. 181–192, 2021.

[58] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and

T. Mikolov, ‘‘FastText.Zip: Compressing text classification models,’’ 2016, *arXiv:1612.03651*.

[59] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ‘‘BERT: Pre-training of deep bidirectional transformers for language understanding,’’ 2018, *arXiv:1810.04805*.

[60] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, ‘‘Distributed representations of words and phrases and their compositionality,’’ in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[61] J. Sivic and A. Zisserman, ‘‘Video Google: A text retrieval approach to object matching in videos,’’ in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 3. Washington, DC, USA: IEEE Computer Society, Oct. 2003, pp. 1470–1477.

[62] Y. Zhu, Y. Zuo, and T. Li, ‘‘Modeling of ship fuel consumption based on multisource and heterogeneous data: Case study of passenger ship,’’ *J. Mar. Sci. Eng.*, vol. 9, no. 3, p. 273, Mar. 2021.

[63] A. Géron, *Hands-On Machine Learning With Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 1st ed. Sebastopol, CA, USA: O’Reilly Media, Mar. 2017, pp. 355–357.

[64] J. J. Hopfield, ‘‘Neural networks and physical systems with emergent collective computational abilities,’’ *Proc. Nat. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, 1982. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554

[65] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, ‘‘Empirical evaluation of gated recurrent neural networks on sequence modeling,’’ 2014, *arXiv:1412.3555*.

[66] S. Hochreiter and J. Schmidhuber, ‘‘Long short-term memory,’’ *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[67] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, ‘‘Tree-structured regional CNN-LSTM model for dimensional sentiment analysis,’’ *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 581–591, 2020.

[68] A. Yenter and A. Verma, ‘‘Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis,’’ in *Proc. IEEE 8th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2017, pp. 540–546.

[69] F. Wei and T. Nguyen, ‘‘A lightweight deep neural model for SMS spam detection,’’ in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Oct. 2020, pp. 1–6.

[70] H. Yang, Q. Liu, S. Zhou, and Y. Luo, ‘‘A spam filtering method based on multi-modal fusion,’’ *Appl. Sci.*, vol. 9, no. 6, p. 1152, Mar. 2019.

[71] K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Spatial pyramid pooling in deep convolutional networks for visual recognition,’’ *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[72] L. Yan, R. H. Dodier, M. Mozer, and R. H. Wolniewicz, ‘‘Optimizing classifier performance via an approximation to the Wilcoxon-MannWhitney statistic,’’ in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 848–855.

[73] H. H. Addeen, Y. Xiao, J. Li, and M. Guizani, ‘‘A survey of cyber-physical attacks and detection methods in smart water distribution systems,’’ *IEEE Access*, vol. 9, pp. 99905–99921, 2021.

[74] J. Ya, T. Liu, P. Zhang, J. Shi, L. Guo, and Z. Gu, ‘‘NeuralAS: Deep wordbased spoofed URLs detection against strong similar samples,’’ in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–7.

[75] J. Song, R. Paul, J. Yun, H. Kim, and Y. Choi, ‘‘CNN-based anomaly detection for packet payloads of industrial control system,’’ *Int. J. Sens. Netw.*, vol. 36, no. 1, pp. 36–49, 2021.

[76] M. Sameen, K. Han, and S. O. Hwang, ‘‘PhishHaven—An efficient real-time AI phishing URLs detection system,’’ *IEEE Access*, vol. 8, pp. 83425–83443, 2020.

[77] Y. Huang, Q. Yang, J. Qin, and W. Wen, ‘‘Phishing URL detection via CNN and attention-based hierarchical RNN,’’ in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2019, pp. 112–119.

[78] S.-J. Bu and S.-B. Cho, ‘‘Deep character-level anomaly detection based on a convolutional autoencoder for zero-day phishing URL detection,’’ *Electronics*, vol. 10, no. 12, p. 1492, Jun. 2021.