Identifying Phishing Mail by Using URL and Mail Features Using ML Based Classification Tools

A project report submitted in partial fulfillment of the requirement

for the award of the degree of

Master of Computer Applications

by

Vipin Singh Negi

(Regd. No.-2018PGCACA58)

Under the supervision of

Dr. Alekha Kumar Mishra

(Internal Supervisor)

NIT Jamshedpur

Department of Computer Applications

National Institute of Technology Jamshedpur (INDIA)

DECLARATION

I hereby declare that dissertation of the work titled, **"Identifying Phishing Mail by Using URL and Mail Features and ML Based Classification Tools"**, submitted towards requirements of project work for partial fulfillment of Master in Computer Applications is an original work of mine and the report has not formed the basis for the award of any other degree, associateship, fellowship or similar titles. I also declare that wherever I have used materials such as data, theoretical analysis, and text from other sources, I have given due credit to them by citing source of the work in the thesis.

Vipin Singh Negi

Regd. No: 2018PGCACA58

Dept. of Computer Applications

NIT Jamshedpur

Place: Kanpur, Uttar Pradesh

Date: 07-06-2021

राष्ट्रीय प्रौद्योगिकी संस्थान जमशेदपुर
NATIONAL INSTITUTE OF TECHNOLOGY JAMSHEDPUR
(An Institution of National Importance under MHRD, Government of India)
Department of Computer Applications

CERTIFICATE

This is to certify that the thesis titled " **Identifying Phishing Mail by Using URL and Mail Features and ML Based Classification Tools**", submitted by **Vipin Singh Negi (Reg. No.: 2018PGCACA58)** towards partial fulfillment of the requirements for the award of degree of Master of Computer Applications, is a bonafide work carried out under the supervision and guidance of mine.

Dr. Alekha Kumar Mishra

(Supervisor)

ACKNOWLEDGEMENT

It gives me immense pleasure to express my deep sense of gratitude to my supervisor Dr. Alekha Kumar Mishra for his valuable guidance, motivation, constant inspiration and above all for their ever-cooperating attitude that enable me in bringing up this thesis in the present form.

My heartfelt gratitude also goes to Dr. D. K. Shaw, Head of Department of Computer Applications for providing me the opportunity to avail the excellent facilities and infrastructure. I am equally thankful to all other faculty members and non-teaching staffs of Computer Applications Department for their guidance and support.

I am also thankful to all my family members whose love, affection, blessings and patience encouraged me to carry out this thesis successfully. I also extend my gratitude to all my friends for their cooperation.

I thank Almighty God, my lord for giving me the will power and strength to make it happen. Lastly, I thank myself for putting for sheer hard work, dedication and perseverance.

Vipin Singh Negi

ABSTRACT

Emails are widely used as a means of communication for personal and professional use. The information exchanged over mails is often sensitive and confidential such as banking information, credit reports, login details etc. This makes them valuable to cyber criminals who can use the information for malicious purposes. Phishing is a strategy used by fraudsters to obtain sensitive information from people by pretending to be from recognized sources. In a phished email, the sender can convince you to provide personal information under false pretenses. This project work considers the detection of a phished email as a classification problem and this paper describes the use of machine learning algorithms to classify emails as phished or ham.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
SVC	Support Vector Classifier
RF	Random Forest
MCC	Mathews Correlation Coefficient
ROC	Receiver Operator Characteristic
AUC	Area Under Curve
APWG	Anti-Phishing Working Group
PCA	Principal Component Analysis
t-SNE	t-distributed Stochastic Neighbor Embedding
SVM	Support Vector Machine
GB	Gradient Boosting Classifier

Chapter 1

INTRODUCTION

1.1 Introduction

Phishing is a type of social engineering attack [2] often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message. The recipient is then tricked into clicking a malicious link, which can lead to the installation of malware, the freezing of the system as part of a ransomware attack or the revealing of sensitive information.

An attack can have devastating results. For individuals, this includes unauthorized purchases, the stealing of funds, or identify theft.

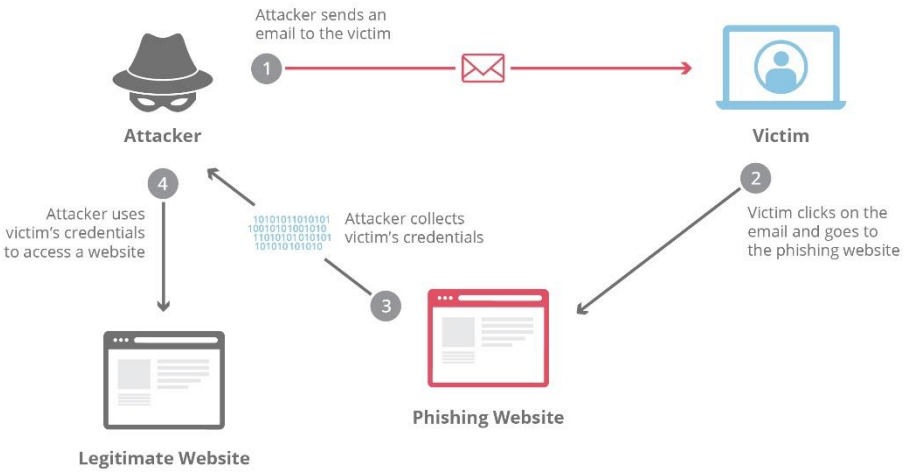


Figure 1: Overview of phishing attack

Phishers rely on two techniques to achieve their goals; they either use the deceptive phishing method or the malware-based phishing (Figure 1). The first technique relies on

social-engineering schemes by using emails to send deceptive links as these emails look a lot like coming from a real business or bank account, and direct the receiver to an affiliated fake website asking to fill in some required details that are confidential such as; usernames, passwords, credit card numbers, and personal information.

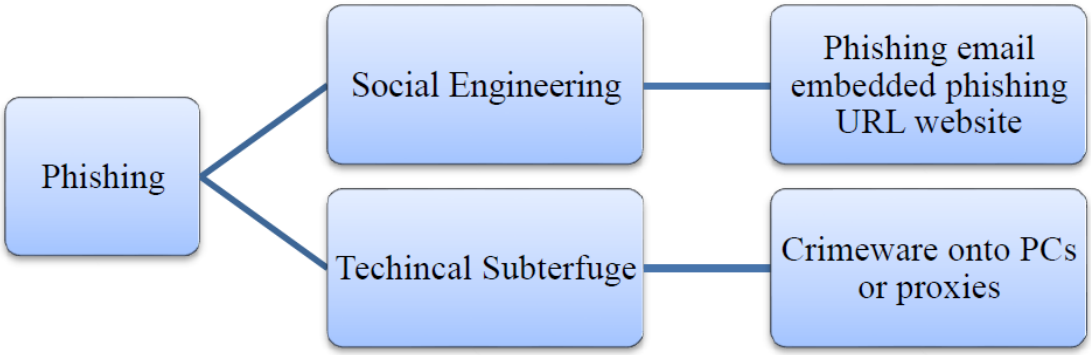


Figure 2: Types of Phishing Emails

While the malware-based phishing technique does not directly ask for details, but it rather relies on malicious codes or malware and technical schemes if users click on the embedded link, or looks for security gaps in the receivers' devices to obtain their online account information directly. Sometimes, the phisher will attempt to misdirect the user to a fake website or a legitimate one monitored by substitutions.

Phishing attacks have reached unprecedented levels especially with emerging technologies such as mobile and social media [13]. For instance, from 2017 to 2020, phishing attacks have increased from 72 to 86% among businesses in the United Kingdom in which a large proportion of the attacks are originated from social media.

The APWG Phishing Activity Trends Report analyzes and measures the evolution, proliferation, and propagation of phishing attacks reported to the APWG. Figure 3 shows the growth in phishing attacks from 2015 to 2020 by quarters based on APWG annual reports [38].

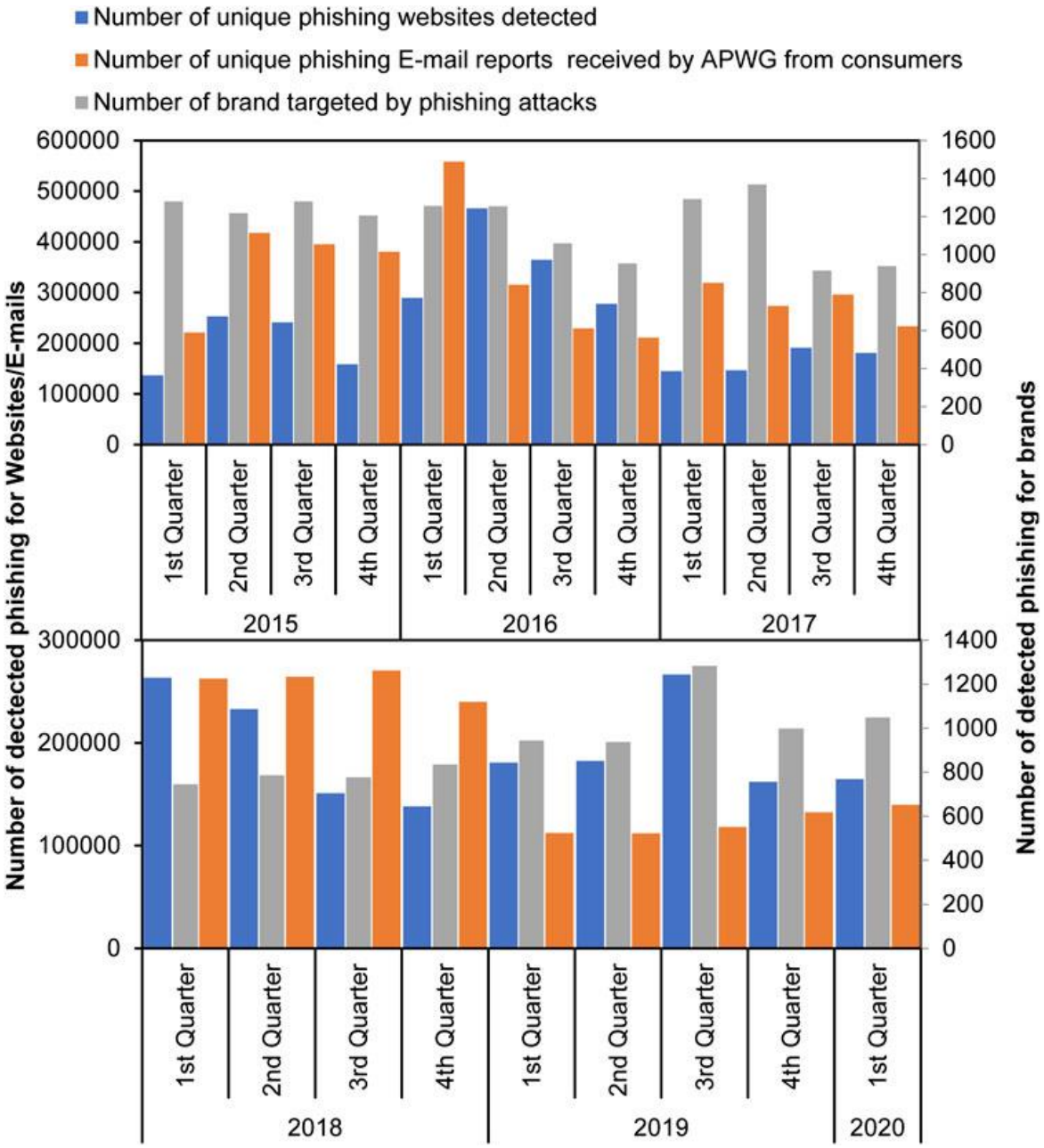


Figure 3: Growth in Phishing attacks from 2015 to 2020

A report from Microsoft (Microsoft, 2020) showed that cyber-attacks related to COVID-19 had spiked to an unprecedented level in March, most of these scams are fake COVID-19 websites according to security company RiskIQ (RISKIQ, 2020).

A study (KeepnetLABS, 2018) confirmed that more than 91% of system breaches are caused by attacks initiated by email. Cybercriminals use email as the main medium for leveraging their attacks.

As shown in the figure 4, online stores were at the top of the targeted list (18.12%) followed by global Internet portals (16.44%) and social networks in third place (13.07%) (Kaspersky, 2020). While the most impersonated brands overall for the first quarter of 2020 were Apple, Netflix, Yahoo, WhatsApp, PayPal, Chase, Facebook, Microsoft eBay, and Amazon (Checkpoint, 2020).

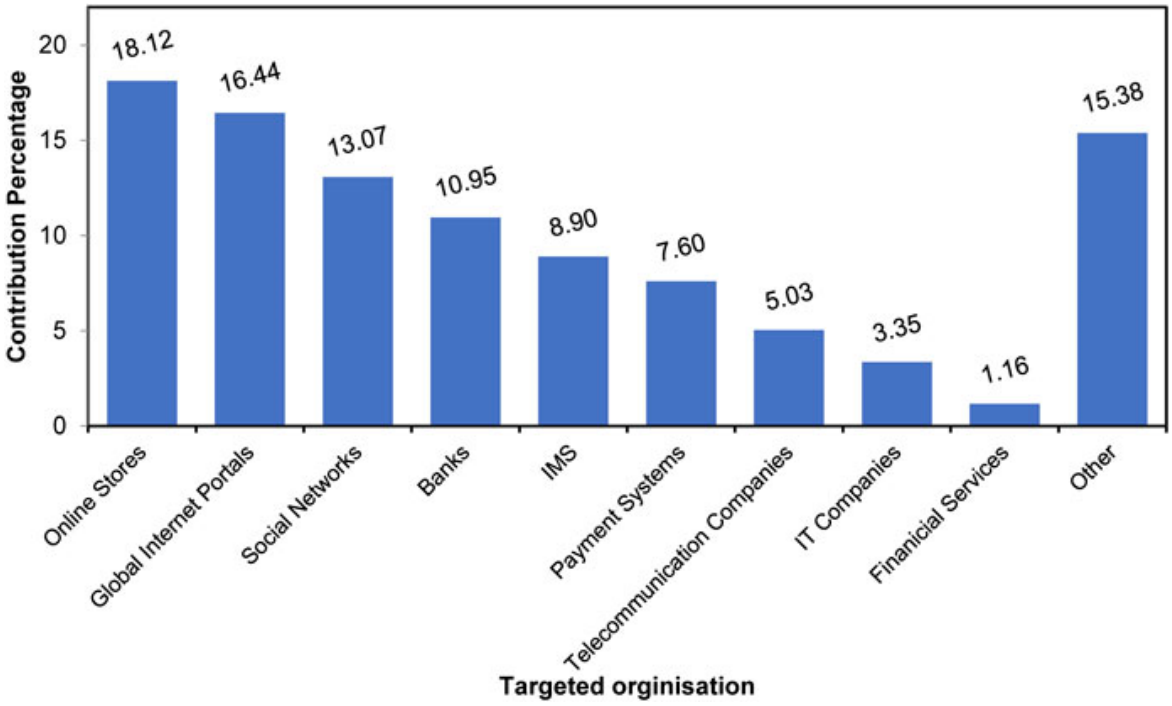


Figure 4: Phishing attack targeted organization

1.2 Problem Statement

Phishing is a social engineering tactic [2] designed to trick users into divulging sensitive personal information, such as one's social security or bank account numbers, through impersonation of a trustworthy third party.

Phishing emails considers as the fastest rising online crime method used for stealing personal financial data and perpetrating identity theft. Individuals who respond to

phishing e-mails, and input the requested financial or personal information into e-mails, websites, or pop-up windows put themselves and their institutions at risk.

With the massive work exists for phishing email detection task, there is no set of features that has been determined as the best to detected phishing. Moreover, the same nondeterministic scenario is applied for the underling classification algorithm. Finally, there is a need to keep on enhancing the accuracy of the detection techniques. Overall the problems carried out in this project are as following:

- How to determine the best set of features to be used with phishing detection
- How to select the best classification algorithm to be used for phishing detection.
- How to enhance the performance of the best selected features and classifiers.
- How to integrate multiple classification algorithms for phishing detection and to evaluate such integration.

1.3 Motivation

The harmful effects of phishing attack can wreak havoc over individual or organization's finances and privacy. In many cases, on successful phishing attack, phishers have even assaulted the privacy of the victim and prevent them from accessing their own account.

Phishing attack are almost 11 times more prevalent than in 2006 and are still the most common cybercrime in 2020. Whooping 96% of phishing attack are from emails and around 75% of organization have experienced some kind of phishing attack in 2020. The information obtained by silently acquiring access to victim's account is also sold in dark net. Therefore, in this study, I will quantify and qualify the phishing email features to prevent and mitigate the risk of phishing email

1.4 Objective

The goal of this project is to conduct a comparative assessment between various classification algorithms techniques, and various features. Moreover, the goal includes the development of multi-classifier integration model by combining more than one classification technique to enhance detection and protecting against phishing emails.

The objectives of this project are as follows:

- Determine and evaluate the best set of features to be used for phishing E-mails detection using manual feature selection based on the email structure.
- To determine the best classification algorithm for phishing detection.
- To fine tune the classification algorithm for best performance.
- Design a system which integrate multiple classification algorithms for phishing emails detection and to evaluate such integration.
- Compare the performance metrics of all the models.

1.5 Scope and Limitation

The scope of this project is phishing emails detection, where features are extracted from the mails. Moreover, Naive Bayes, Random Forest, Logistic Regression etc. top classification ML algorithms were used for phishing emails detection. This project also target to develop an integration of best performing classifier for better prediction.

For the limitation, this project will not cover the phishing websites, moreover the experiments will not cover all the available classification algorithms. This project also doesn't take spams into the consideration even though phishing and spam are unwanted yet they are fundamentally different. However, this study will evaluate experimentally the most well-known algorithms.

1.6 Outline of Chapters

The thesis is consists of five chapters organized as the follows:

- **Chapter One:** (Introduction) It gives an overview of phishing detection techniques, problem statement, the objective of the study, the motivation, the scope and limitation, thesis contribution and finally outline of chapters
- **Chapter Two:** (Literature review) this chapter provides an overview of the related works in phishing emails detection and summary of articles that published by other researchers.
- **Chapter Three:** (Methodology) this chapter provides an outline of the project methodology which used in this thesis. Overview of the software that used for the evaluation of the proposed method and the dataset were used in this project.

- **Chapter Four**: (Implementation) this chapter describes the implementation details of experiment and the results that were obtained for all the proposed scenarios and comparison of the results.
- **Chapter Five**: Conclusion and future work.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

Detection of phishing emails has received a lot of attention recently due to their impact on users' security. Therefore, many techniques have been developed to detect phishing emails varying from communication-oriented techniques, such as authentication protocols, blacklisting, and white-listing, to content-based filtering techniques. The blacklisting and white-listing techniques have not proven though to be sufficiently efficient when used in different domains, and thus they are not commonly used. Meanwhile, the content-based phishing filters have been widely used and have proven to be of high efficiency. In light of this, researches have focused on content-based mechanism and on developing machine learning and data mining techniques based on the header and body of emails.

In 2007, a study was conducted to measure the efficiency of the existing tools for phishing detection. This study showed that even the best phishing detection toolbars missed over 20% of the phishing websites [14]. Another study, which was conducted in 2009 concluded that most anti-phishing tools did not start blocking phishing sites before several hours or days have passed after these phishing emails sent luring users [15]. Therefore, I conclude that the currently implemented detection tools do not detect these phishing email and websites completely (100% percent) [16].

This Chapter presents different Algorithms for the detection and prediction of phishing emails.

2.2 Phishing E-mail Detection Techniques

A wide range of filters have been developed by specialists to predict and prevent phishing emails and manage occurring threats relying on either traditional techniques such as authentication protection [8], or on modern techniques of learning machines or mining data. The phishing e-mail detection is broadly divided into two categories:

i. Traditional Methods
ii. Automated Methods

2.2.1 Traditional Methods

Traditional methods of detection fall into two categories, the network-level protection and the authentication protection. The first category of protection at a network level includes blacklist filters and white-list filters which prevent phishing by blocking suspected IP addresses or domains from accessing the network. In addition, there are the Pattern Matching filters and the Rule-based filters which rely on manually entered and updated fixed rules for detection.

- **WHITELIST FILTER**

White-list filtering provides protection at network level as well, but in contrary to blacklists; this technique compares the email's data with a pre-defined list containing static IP addresses of legitimate domains and IP addresses [17]. In this regard, only emails with data matching the list will be allowed to access the network to the user's inbox.

Email addresses and IP addresses are included in the white-list if they belong to legitimate users or companies who have agreed to add their addresses to this list. Emails with data matching to this list will only be classified as legitimate based on this filter, while other emails are considered phishing and prevented from accessing the network for which this filter in called also legitimate emails classifier.

- **PATTERN MATCHING FILTER**

The pattern matching technique filters emails based on specified patterns, including words, text strings, and character sets mentioned in the email's content, subject, or sender. The filter searches through the email for these specified patterns to classify the email into

phishing or legitimate. Although this technique provides protection at a network level, it still provides some invaluable and false results due to the huge number of received emails which may include banned words or text strings but shall not be prevented.

The second category, authentication protection, provides security on both user and domain levels. For a user-level protection, users will have to provide authentications before sending their messages such as verified email and password, while the authentication protection on a domain-level is created for emails servers.

- **EMAIL VERIFICATIONS**

Email verification is a user-level authentication method that requires verification from the sender and the receiver. Once the sender accepts the notification message, the email will be certified and classified as legitimate to be passed into the receiver's inbox [4]. Otherwise, the email will be considered as phishing and thus prevented from accessing the inbox [19].

This filter has its pros and cons. Although this filtering process has proven to be efficient in detecting phishing emails completely (100%), it still needs a lot of time relatively as the receiver has to respond before receiving the message, and there is a risk of losing the email if the verification process generated traffic over the network or the same challenge has not been recognized.

- **PASSWORD FILTER**

Password filters also provide protection through a user-level authentication. Using this filter allows for receiving any email in the subject line, the email address, the header field, or in any part of the email only if the filter was able to detect the determined password. Therefore, if the filer was not able to find the password or detect a wrong password, the email will be rejected. These passwords are not created by default, therefore; first-time users of this filter will have to start a conversation with each other to set and activate a password and then be classified as legitimate by the filter. This type of filters still has its shortcoming in terms that some legitimate emails might be lost if the password was not recognized, in addition that the process requires time.

2.2.2 Automated Methods

This method applies automated classifiers that rely on machine learning and data mining. These classifiers work beside the server and filter the received emails into phishing or legitimate by examining different features if the email's header and body [20].

- **LOGISTIC REGRESSION**

The logistic regression is a widely-used method due to its easily-interpretable and practical results. This model is functional in predicting binary data (0/1 response) as it relies on statistical data and applies a generalized linear model [7].

Despite of this method's simplicity, it has three shortcomings; first, it requires more statistical assumptions before being applied. Second, it its more functional with variables that have linear relation than those with a complex relation. Last, the accurateness of the predication rate is sensitive to the completeness of the data [20].

- **CLASSIFICATION AND REGRESSION TREES (CART)**

The Classification and Regression Trees (CART) model developed in 80's is used to represent the distribution of Tree that splits using two components, and the T tree that splits into two nodes Decision trees are represented by a set of Yes or No questions which splits the learning sample into smaller and smaller parts.

Unlike logistic regression method, this model is used for complex relations between variables rather than linear relations

A binary tree is created by continuously partitioning the predictor space into different homogenous groups. The partition occurs depending on defined splitting rules associated

to the internal nodes of the tree, where each homogenous group is associated by a terminal node.

This model leads to generating a big binary tree which, although is practical for complex relations and provides easily-read interactions among predictors; it still makes it hard to predict the additive effects due to its huge [21].

- **DECISION TREES FILTER (DT)**

Decision Trees Filter is a graphical model of classification that is comprised of nodes and arrows. The base node is called the Root from which the DT is initiated. Each node within the network contains an “If-then” rule, a class, and a feature, and leads to the next one using the arrows, referred to as edges. The decision tree ends with a leaf node called the terminator. The tree could include one or more classifier stages and the internal nodes are bounded by the root and terminating nodes.

Different algorithms have been suggested to generate decision trees including the ID3 model which calculates information of entropy as a heuristic function to evaluate the target. In 1992, this algorithm was developed to C4.5 algorithm.

In that sense, the decision tree will generate sub-trees, each node in the tree has a parent node leading to it (except for the root), and each one also leads to a child node (except for the terminating node), while the tree will end with the terminating node (leaf node) that represents the final solution of the suggested problem.

- **SUPPORT VECTOR MACHINE (SVM)**

SVM is widely applied by researchers in the medical diagnoses, text categorization, image classification, bio sequences analysis, and other fields. Using this technique, data is divided into two categories using statistics, Quadratic equations, and fixed rules. The binary classification of the data is created by using a separating hyper plane to maximize the space of the margin base on kernel functions, and extracting data and storing it in the vector, to reach the best solution of the problem and finding the suitable classification. This technique is beneficial for finding solutions of problems with unfamiliar history, but fails at analyzing big data.

Table 1.0.1 summarizes the well-known phishing detection tools such as CloudMark, Netcraft, FirePhish, eBay Account Guard and IE Phishing Filter. The authors pointed out the main disadvantages of the popular tools that are widely used.

Tool	Type	Description	Advantage	Disadvantage
Snort	Network level	Heuristic tool	Good at detecting level attacks	Rules require manual adjustments. Does not look at content
Spam Assassin	Server Side Filter	Heuristic engine uses specific features	Good at detecting email header spoofing	High false positives
PILFER	Server Side Filter	Utilize 10 features	Better performance than spam assassin	Did not use content from the body of the email. Used with short lived phish domains.
Spoof Guard	Client Side Tool	Plug-in to a browser	Warns user if link points to phishing site	Users do not pay attention to warnings. Not all email clients are browser based.
Calling ID, Cloud Mark, Netcraft, and Fire Phish	Client Side Tool	Utilizes blacklist of domains	Good for domains that employ domain level authentication	Phish domains are short lived. Does not look at email content.
eBay Account Guard	Client Side Tool	Utilizes blacklist of eBay URLs	Protects eBay users.	Specific website tool.
IE Phishing Filter	Client Side Tool	Records specific user website visiting patterns.	Adapts to user website visit pattern	Works only on internet explorer.
Catching Phish	Client Side Tool	Detects fake website based on rendered images	Browser independent. Good results on small data sets.	Processing time is high. Susceptible to screen resolution

Table 1: Phishing Detection Tools

2.3 Literature Review

This section provides an overview on some of the main studies conducted on data mining techniques and algorithms to detect phishing emails:

Chandrasekaran depended on the distinctive structural features of the email to detect phishing emails. These features work in cooperation with the SVM to predict phishing emails and prevent them from originally reaching the user[18].

In 2007, Abu-Nimeh[20] focused on examining different machine learning methods and comparing the accuracy of their predictions using a total of 2889 phishing and legitimate emails. All of the following methods were included in the study: Logistic Regression (LR), Classification and Regression Trees (CART), Bayesian Additive Regression Trees (BART), Support Vector Machines (SVM), Random Forests (RF), and Neural Networks (NNet) which were tested also using 43 features. According to the results, RF outperforms all other classifiers, with an error rate of 07.72%, followed by CART, LR, BART, SVM, and NNet respectively, providing that the legitimate and phishing emails are given equal weights. In terms of false positive rate, the best results were achieved by the LR with a

percentage of 4.89%, followed by BART, NNet, CART, and SVM respectively, while the worst false positive rate was achieved by the RF with percentage of 08.29 [20].

Furthermore, the two methods of adaptive Dynamic Markov Chains (DMC) and latent Class-Topic Models (CLTOM) were proposed by Bergholz to classify emails where two new features were produced. The adaptive version of the DMC succeeded to provide the same quality performance in comparison to the standard version while using two-thirds less of the memory. As for the CLTOM, the adaptive version has shown higher performance than the standard LDA as the first incorporates class-specific information into the topic model and has achieved a total of topic numbers of up to 100 [22].

Toolan developed a new C5.0 algorithm to filter into Phishing / non-Phishing categories by selecting 5 features. The sampled data included 8,000 emails where half of them were phishing and the other half was legitimate. This approach outperformed any other individual classifier or collection of classifiers in terms that is achieved higher recall efficiency [23].

Abu-Nimeh and others founded a detection tool for protecting mobile platforms against attacks. The client-server distributed server relies on Additive Regression Trees beside the server with the assistance of the automatic variable selection to improve their predictive accuracy and eliminate the overhead of variable selection is applied [20].

Gansterer proposed a filtering system that classifies received emails into three categories; legitimate (solicited e-mail), spam, and phishing emails, relying on newly developed features from these emails. The system comprises different classifiers to be able to categorize received messages. A classification accuracy of 97% was achieved among the three groups, which is considered better than solving the ternary classification problem by a sequence of two binary classifiers [24].

Dr. Ma used an algorithm with a set of orthographic features to cluster phishing emails automatically and eliminating redundant features. This clustering and feature selection technique succeeded in providing highly efficient results. Ma applied the global k-mean model with a little modification and generated the values of the objective function over a range of tolerance values of selected features subsets. The objective function values assisted in recognizing the suitable clusters based on the distribution of these values [25].

Basnet studies a detection approach that utilizes readily acquired features from the email's content without resorting to heuristic-based phishing features. This approach relied on Confidence-Weighted Linear Classifiers proposed by Basnet. images are generated by Phishers from the message's text that only graphical data passes the phishing filter [26].

Dr.Wu focused on spoofing emails and Microsoft OutlookTM services by developing a sender authentication protocol (SAP). This authentication protocol verifies the authenticity of the sender by testing the claimed-sender1 with the archived emails. The enhanced OutlookTM has an add-in that tests feasibility while it remained the same user-friendly interface of the original version, and this the SAP add-in will be started automatically once the OutlookTM operates [28].

In 2011 Khonji and Jones and Iraqi they listed the 47 features for the Email that were used to classify the phishing emails in the study and they gave a brief description on each feature, the list covers all the structures of the Email [29].

A new genetic algorithm was developed by Alguliev for clustering spam messages and solving clustering problems. The proposed algorithm uses the strategy of maximizing the similarity between messages in clusters, and the objective function is defined by k-nearest neighbor algorithm. However, such algorithms are limited by the constant support of chromosomes which reduces convergence process when trying to solve constrained problems. Therefore, a penalty function is applied to expedite the convergence process and preventing infeasible chromosomes

Thereafter, a detailed examination is conducted on the resulting classification to conclude information about the classes, and an informative portrait is shaped through documentation to achieve better understanding of these clusters and spam messages. This anti-spam system will help in predicting targeting information attacks, in addition to analyzing the origins of spam messages which will help in finding organized social networks of spammers [29].

Azad has focused on testing different existing algorithms in terms of their accuracy, such as Naive Bayes, logistic regression, and support vector machine (SVM) classifiers. He used bag of words and augmented bag of words models. In general, the tested classifiers achieved high results indicating an accuracy rate of 95% with the SVM with the linear

kernel and Bayes topping the other classifiers, as they only missed 10 and 2.66 percent of phishing emails respectively. When in comparison with the Naive Bayes and logistic regression, the SVM showed equal results being tested with less features. Meanwhile, the linear SVM was tested as well with removing additional features to result in lower detection rates as it misclassified 5.86 percent of phishing emails, meaning that additional features enhance the accuracy of the results. In conclusion, the study showed that linear SVM is beneficial for detecting phishing emails before they even reach the user's inbox.

A new method for clustering of spam messages collected in antispam system is offered by Alguliev , through the development of Genetic algorithm including penalty function for solving clustering problem. In addition to, the classification of new spam messages coming to the bases of antispam system. The proposed system is not only capable to detect purposeful information attacks but also to analyze origins of the spam messages from collection, it is possible to define and solve the organized social networks of spammers [29].

Meanwhile, a new version of neural networks was developed by Al-Momani that achieved a zero-day detection of unknown phishing emails. The new framework was named PENFF (Phishing Evolving Neural Fuzzy Framework) which relies on adaptive evolving fuzzy neural network (EFNN). As a performance indicator; the Root Mean Square Error (RMSE) and Non-Dimensional Error Index (NDEI) are 0.12 and 0.21 respectively which indicate low error rates compared to other approaches [30].

Kumar used TANAGRA data mining tool on a sampled spam dataset to evaluate the efficiency of the emails classifier where several algorithms were applied on that data set.

At the end, the features selections by Fisher spam filters and Rnd filtering achieved better classifications. After fisher filtering has acheived more than 99% accuracy in detecting spam, The Rnd tree classification algorithm was applied on relevant features. [31]

Altaher relied on Adoptive Evolving Fuzzy Neural Network (EFuNN) to create Phishing Evolving Neural Fuzzy Framework (PENFF) to detect of unknown "zero-day" phishing emails by handling all similar feature vectors to establish rules for prediction. Therefore, PENFF approach relies on the similarity of features included in the email's body and URL [30].

Pandey classified phishing emails by applying several methods, such as; Multilayer Perceptron (MLP), Decision Trees (DT), Support Vector Machine (SVM), Group Method of Data Handling (GMDH), Probabilistic Neural Net (PNN), Genetic Programming (GP) and Logistic Regression (LR). This combination aimed at using text and data mining in parallel for detection where 23 keywords were extracted from the email body and were already included sampled data set, in addition to a total 2500 phishing and non-phishing emails were analyzed.

Jameel and George used a feedforward neural network to identify the phishing email by extracting features from the email's header and HTML body. Their suggested algorithm was tested on 18 features using 5 hidden neurons. For this algorithm, a training is required before implementing it which takes 173.55 msec. The time for testing a single email is 0.00069 msec. The consumed time will increase with the increase in the neurons number while it is still considered low. With regard to the results, the algorithm proved high accuracy of 98.72%, and a learning rate of 0.01. [33]

Zhang aimed at estimating the accuracy of the cross validation approach in detecting phishing emails. He used multilayer feedforward neural networks (NN) systems with different numbers of hidden units and activation functions to prove that NNs can provide fairly accurate and efficient results with an estimated number of hidden units. It is worth mentioning that he proved these results even with few training while selecting the features set will achieve better results [33].

In 2013, Al Momani found a new model that proved excellent results in terms of true positive, true negative, sensitivity, precision, F-measure and overall accuracy compared with other approaches. In addition, the system showed efficiency in predicting the values of these emails in online mode, and long-life working with footprint consuming memory. The model Al Momani developed is called Phishing Dynamic Evolving Neural Fuzzy Framework (PDENF) for predicting unknown phishing emails and detecting them in zero day [30].

Regarding websites classification, Khonji examined the modified technique for preventing phishing emails and enhancing the filters efficiency. The previously proposed technique relied on analyzing the website's URLs lexically which enhanced the accuracy

of the filters by 97%. Lexical URL analysis indicated higher accuracy of anti-phishing classification [28].

Later in 2013, Rathi aimed at comparing the performance between algorithms with a feature selection and algorithms without a feature selection. At first, the sampled data was examined without any filters or features selection, then the classifiers were tested each at time beginning with the best-first feature selection to be able to elect the most beneficial features and then apply various classifiers for classification

The Random Tree classifier proved a 99.72% accuracy which means it works best to detect spam emails. In conclusion, the accuracy of email filters was enhanced incredibly when the algorithm with feature selection was applied into the entire process and that classifiers of tree shape are more efficient in detecting spam emails [34]

Another framework was found by Al Momani and others that also detects unknown zero-day phishing emails relying on a the "evolving connectionist system". The new system was named the phishing dynamic evolving neural fuzzy framework (PDENFF) and follows a hybrid learning approach (supervised/ unsupervised) and is supported by an offline learning feature to achieve the intended purpose. Using this system helped in enhancing the detection of zero-day phishing e-mails was improved between 3% and 13%. Moreover, it used rules, classes or features to enhance the learning process using ECOS which provided the system with the advantage of distinguishing phishing emails from legitimate one [30].

Another mechanism was developed later in 2014 by Akinyelu to better classify phishing emails using forest machine learning mechanism. This mechanism was tested on data comprising around 2000 phishing emails with advanced features (as identified from the literature), and it was able to classify phishing emails with high efficiency (99.7%) with low false negative (FN) and false positive (FP) rates. Therefore, Akinyelu's algorithm is more efficient in terms that it requires fewer features to detect phishing and provides more accurate results [35].

A fraudulent detection model was proposed by Nizamani (2014) using an advanced selection of features where the different categories were compared in terms of the fraudulent email detection rate. The study was conducted applying several classification

approaches and algorithms, such as SVM, NB, J48 and CCM, in addition to different features sets. An accuracy percentage of 96% was achieved and the results indicated that the level of accuracy was affected by the type of selected features rather than the classifiers' type [36].

In 2020, Meenakshi Das, Somya, Alekha Kumar Mishra presented a paper that provides comparative study of most of the phishing tools and techniques in the present world [6].

In 2020, R.M. Verma & Aassal also released a descriptive paper aiming the benchmarking of the phishing detection research. [10]. Ryan M and Schuetzler presented a paper suggesting the phishing attack in future [9].

In 2015, Kathirvalavakumar and others proposed a multilayer neural network to detect phishing emails. His suggested network relies on a feedforward pruning algorithm that extracts distinguished data and features from the email and applies a weight trimming strategy. This pruning strategy helps in minimizing the number of features through the algorithm resulting in minimum computation required for classification of emails into phishing or not. The network has provided fair results in terms of false positives and false negatives. As this network has been tested on data from 2007, using this network for current data requires identifying the new features to the algorithm incorporating them into input domain for training in order to be useful. [37].

Chapter 3

METHODOLOGY

3.1 Introduction

This chapter presents detailed project work on phishing detection. The work contribute to the field by developing an ensemble integration model by combining top 5 fine-tuned classifications techniques to enhance the detection accuracy. Moreover, a comparative assessment between various classification algorithms is proposed.

3.2 Project Approach

The project work starts by going through the various research paper and investigating the phishing email. Then, determining set of features that are prominent in the phishing mail. The mails are downloaded from various sources and a CSV dataset is constructed by extracting the feature set values from the mails. The dataset is now subjected to various visualization plots and dimensionality reduction techniques. Feature Importance is also plotted to show case the important features in the process of classification. Various classification algorithms are then implemented upon the dataset. Top 4 classification algorithms are selected on the basis of the cross validation error. The selected models are then integrated [11] to form an ensemble model to get better performance. The steps of the proposed work is presented in Figure 5.

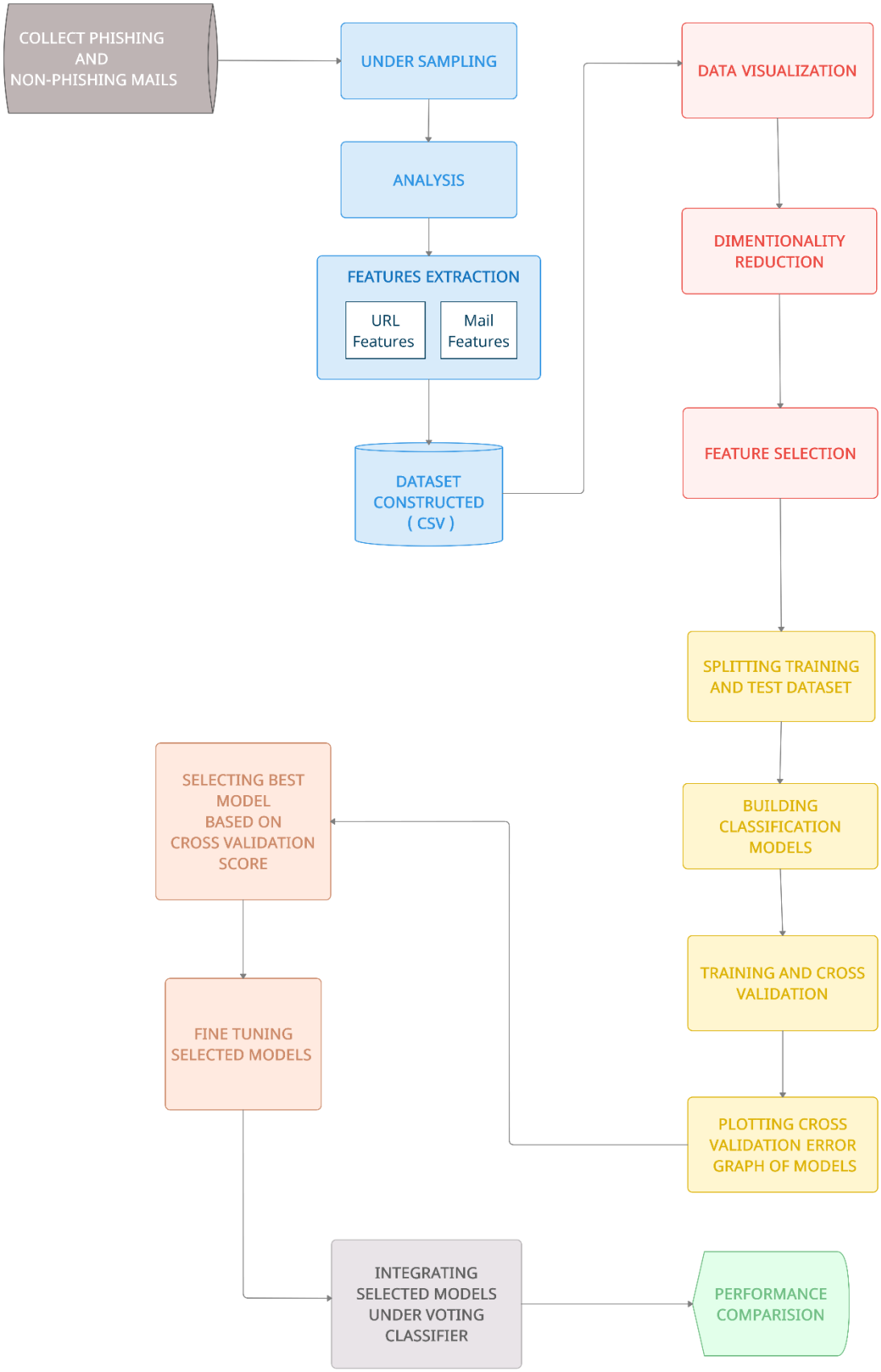


Figure 5: Workflow design of project

3.2.1 Pre-processing

In the pre-processing step, phishing emails dataset is collected. A thorough analysis of ham and phishing mails will done to gain the knowledge of important features to be extracted. Word Cloud Analysis would be very handy to get to know insight of both the type of mails. Features to be extracted will be determined based on the analysis. Features are then extracted from each email and all the features for all emails are presented in a list of array, each row represent one email along with columns corresponding to selected features, in addition to a column that represents the class of the email (whether a phishing or legitimate email) as show in in Table 2. The set of features [12] were categorized into four groups; E-mail body group (contains 9 features), URL features group [3] (contains 9 features), Subject features group (contains 11 features) and Senders address (contains 2 features).

These feature are presented in tables below:

Feature	Description
Html	Binary feature showing if mail has HTML content
Html form	Binary feature showing if mail has HTML < form>
Html iframe	Binary feature showing if mail has HTML < iframe>
Javascript	Binary feature showing if mail has javascript
Flash Content	Binary feature showing if mail has flash content
General Salutation	Binary feature showing if mail has general salutation like dear user, dear customer
Attachments	Number of attachments
Popups	Binary feature showing if mail has popups
Body Richness	Continues feature, ratio of number of words and unique words in body

Table 2: Mail body features

Feature	Description
Number of URLs	Shows number of URLs
Malicious URLs	Shows number of malicious URLs
Text link disparity	Shows if there is a disparity between text and link associated with it.
IP URLs	Shows number of IP URLs
Hexadecimal in URLs	Shows if there is hex encoding in URL
Bad Ranked Domain	Presence of bad ranked domain
Max Domains Counts	Maximum number of domain in a URL
@ in URLs	Shows if any URL has @ in it
Mail to	Shows the presence of mail to: in body

Table 3: URL Features

Feature	Description
Re mail	Shows if mail is reply mail
Fwd mail	Shows if mail is forwarded mail
Contains account	Shows if subject line has account in it
Contains verify	Shows if subject line has verify in it
Contains Update	Shows if subject line has update in it
Contains prime targets	Shows if subject line has prime targets(like amazon,bank,paypal etc) in it

Contains suspended	Shows if subject line has suspended in it
Contains password	Shows if subject line has password in it
Contains urgent	Shows if subject line has urgent in it
Contains access	Shows if subject line has access in it
Subject richness	Shows subject line richness

Table 4: Subject Features

Feature	Description
Number of dash	Shows number of dash in address
Number of dots	Shows number of dots in address

Table 5: Sender Address Features

3.2.2 Data Visualization

The dataset formed from the features extracted from the mails would be then subjected to various visualization. The visualization of data helps in revealing hidden trends in the data which could be proved very effective during training of our model.

We went with plotting the distribution of the features data. Typically, I went for plotting PCA visualization and heatmap.

Furthermore, the dataset was subjected to t-SNE also and I tried to find the trends between class and the feature.

3.2.3 Feature Selection

Feature selection [4], also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. It would be applied to select only the relevant

features for building the model. It is very helpful in reducing the model training time and also improve the accuracy as model is not affected by unnecessary feature variables.

Feature selection is implemented using 4 filters [5]:

i. Univariate filter

ii. Variance filter

iii. High Correlation filter

iv. Feature importance with Random Forest Classifier

The feature importance (variable importance) describes which features are relevant. It can help with better understanding of the solved problem and sometimes lead to model improvements by employing the feature selection.

We have employed Random Forest Classifier for evaluating the feature importance. It fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

3.3 Algorithm Evaluation

We have selected 11 classification algorithms to train and test the accuracy of phishing email classification with the features. The reason behind selecting these algorithms is the different training strategy they use in discovering the rules and the mechanism of learning and testing, the selected algorithms are:

a. Gaussian Naïve Bayes

b. Support Vector Classifier (SVC)

c. Decision Tree Classifier

d. AdaBoost Classifier

e. Random Forest Classifier

f. Extra Trees Classifier

g. Gradient Boosting Classifier

h. Multi-Layer Perceptron

i. K Nearest Neighbors Classifier

j. Logistic Regression

k. Linear Discriminant Analysis

3.4 Voting Ensemble Approach for Phishing Email Detection

We have made voting ensemble classifier [5] on top of five best performing models. The meta-classifier help us to achieve best prediction on the phishing mail classification.

The prediction from above five model are fed into the voting classifier to make the final prediction. Figure 6 shows the best description.

The voting classifier will throw out the final prediction which would also be the best on the basis of votes among various classifiers.

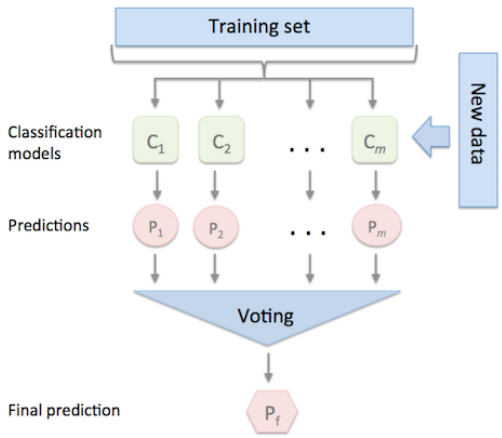


Figure 6: Voting Classifier on top of selected models

Chapter 4

IMPLEMENTATION

4.1 Introduction

This chapter presents the dataset that was constructed from the mails. It further throws lights upon various findings of the experiments and lastly presenting the performance of the classification algorithm.

4.2 Feature Extraction and Dataset Preparation

The utilized dataset contains 4478 emails, 2339 phishing emails and 1650 legitimate emails. The legitimate mails are then oversampled to make balanced database. The emails are obtained from two sources,

For phishing mails: J. Nazario. PhishingCorpus (monkey.org)

For legitimate mails: SpamAssassin PublicCorpus (apache.org)

The spam Assassin resource offers, legitimate emails that contains two categories: easy legitimate emails and hard legitimate emails which are very close to spam then the whole.

Phishing corpus contains phishing mails that are intended for phishing purpose and are completely identical to real life phishing mails.

The mails from both the corpus were analyzed to determine features to be extracted. Word Cloud analysis is also performed on them. The word cloud analysis is performed on subject and body both. The results are shown in figures below.

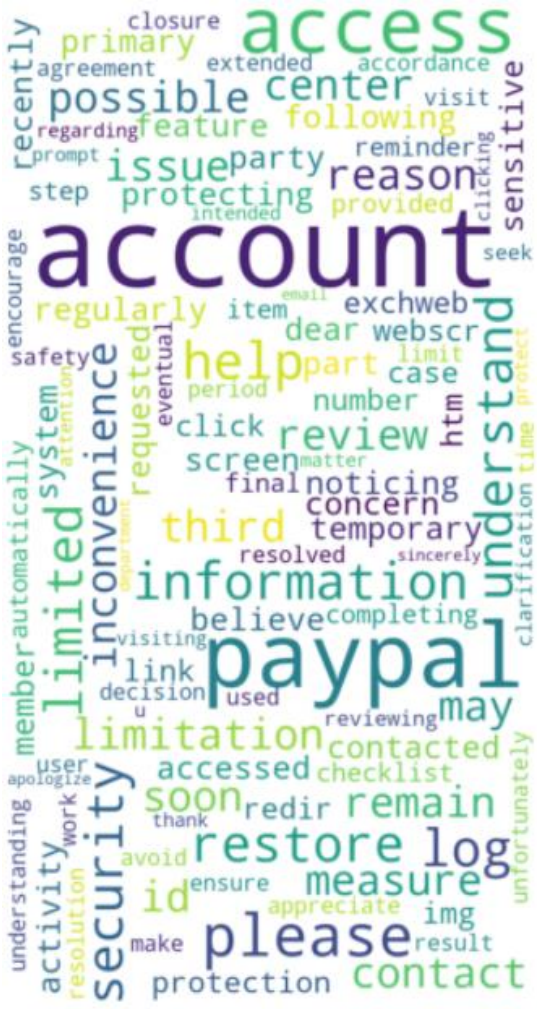


Figure 7: Phishing mails body word cloud

Figure 8: Ham mails body word cloud

Figure 9: Phishing mails subject word cloud word cloud

Figure 10: Ham mails subject

By closely observing the word cloud analysis, we have added some of the crucial features that will help us in phishing mails prediction.

As it could be said that if certain keywords as account, verify, suspended etc. are tracked as feature we can better detect the phishing mail. As per proposed approach we have aimed to include most of crucial features.

Feature extraction is implemented in the data set representation, were each email is converted into feature vector of selected features and a column which represent the type of the email (whether it is phishing or legitimate email) as shown in Figure 11.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
HTML	HTMLForm	IFrame	FlashConte	General Sa	Javascript	mailto:	popups	body richn	Number of	Malicious	text link di	Attachmer	IP URLs	hexadecim	Bad Rank
0	0	0	0	0	0	0	0	1.24	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1.222222	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1.235294	2	0	0	0	0	0	0
0	0	0	0	0	0	1	0	1.927536	3	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1.844037	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0	1.632	11	1	0	0	0	0	0

Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
Maximum	@_in_url	Subject ric	Fwd: mail	Re: mail	contains a	contains v	contains u	contains p	contains s	contains p	contains u	contains a	number of	number of	Class
1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	Ham
1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	Ham
1	0	1.166667	0	0	0	0	0	0	0	0	0	0	2	0	Ham
1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	Ham
1	0	1	0	0	0	0	0	0	0	0	0	0	2	0	Ham
1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	Ham
1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	Ham

Figure 11: Dataset constructed after feature extraction from mails

This project used email and mailparser library of python to parse the mails. All the mail body features are extracted either through above mentioned libraries or through use of regular expressions. URL features are better extracted by use of BeautifulSoup library of python along with the tldextract to extract top level domain and using alexa to fetch ranks of domains.

In subject features, we used jaro distance to see if there is presence of words closer or equal to lists of suspected words. Jaro distance is used as it can overcome the corner case of misspelled words which is one of many characteristics of phishing mail.

The final dataset with extracted features value is prepared and stored in CSV file for usage in coming phase.

4.3 Exploratory Data Analysis

In order to reveal hidden trends in data and to get better prediction at model building phase, we must perform EDA.

4.3.1 Data Cleaning

As dataset is constructed directly from mails by extracting the features value. No null valued cell is found. But several duplicates rows were found which were then dropped from the dataset and the final dataset has:

Number of Ham class records: 1238

Number of Phishing class records: 1135

Dataset shape = (2373 x 32)

4.3.2 Data Visualization

For data visualization I have plotted those using different graphs. They are

a. Heatmap

Heatmap is defined as a graphical representation of data using colors to visualize the value of the matrix. In this, to represent more common values or higher activities brighter colors basically reddish colors are used and to represent less common or activity values, darker colors are preferred. Heatmap is also defined by the name of the shading matrix. Heatmaps in Seaborn can be plotted by using the seaborn.heatmap() function.

As I can interpret from the heatmap shown in Figure 12, there is no such very strong correlation between features but I have decent correlation between:

i) JavaScript and Iframe

ii) Iframe and @_in_url

iii) Text link disparity and maximum domains counts

iv) IP URLs and Number of URLs

v) Malicious URL and Number of URLs

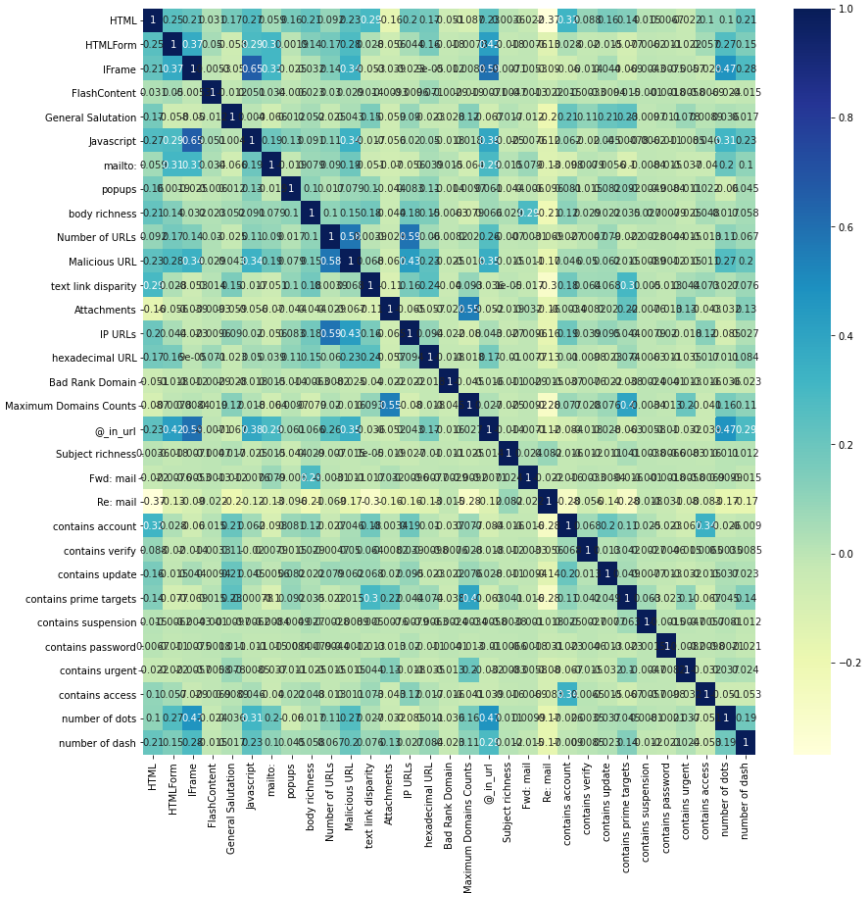


Figure 12: Heatmap

These correlations are quite obvious trend revealed in the heatmap. As number of URLs increases the mail tend to be more malicious and have IP URLs. It also reveals that if a mail has many domains it is much likely to have text link disparity.

b. PCA visualization

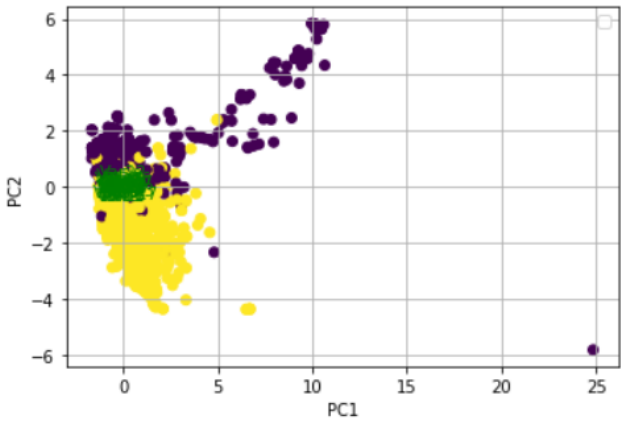


Figure 13: PCA Visualization

PCA is also used for visualization. It is a technique used to emphasize variation and bring out strong patterns in a dataset. It's often used to make data easy to explore and visualize.

Here, I can see that clusters are forming which means there is a separation between the both classes but I can also see the green points which represent mixing presence of both classes and it means often time phishing mail are most like the ham mails.

c. PCA with Singular Value Decomposition

Singular Value Decomposition and its contribution in molding up Principal Component Analysis which is a sophisticated method of extracting important features of a Data Matrix in Machine Learning. But in our case no such new information has been revealed from this visualization it is as same as the PCA visualization.

PCA with SVD also shows the formation of cluster of phishing mail which also overlaps many hams mails as it covers that variance in features too. PCA with SVD is show in Figure 14.

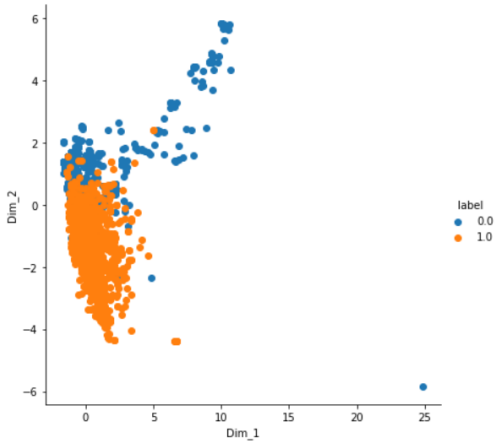
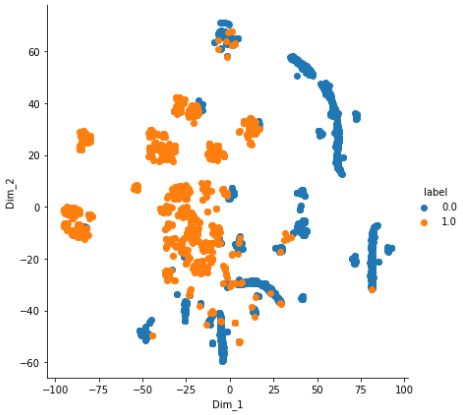


Figure 14: PCA with SVD

d. t-SNE visualization

This is the best t-SNE visualization we get by running t-SNE at perplexit = 35 and number of iteration = 3500. We can see a phishing mails cluster forming in orange color along with that we see several outliers which shows how phishing mails are so close to the ham mails.



4.3.3 Feature Selection

Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in. Having irrelevant features in your data can decrease the accuracy of the models and make your model learn based on irrelevant features.

We went on implementing 4 filters to get most important features. These are:

i) Variance Filter

VarianceThreshold() is a simple baseline approach to feature selection. It removes all features whose variance doesn't meet some threshold. By default, it removes all zero-variance features, i.e. features that have the same value in all samples.

Following features are left after filteration:

```
Index(['HTML', 'body richness', 'Number of URLs', 'Malicious URL',
       'text link disparity', 'IP URLs', 'hexadecimal URL',
       'Maximum Domains Counts', 'Re: mail', 'number of dots',
       'number of dash'],
      dtype='object')
```

Figure 15: Variance filtered features

ii) Univariate filter

Univariate feature selection works by selecting the best features based on univariate statistical tests. It can be seen as a preprocessing step to an estimator. I have used SelectKBest which removes all but the k highest scoring features. Filtered features are:

```
Index(['HTML', 'General Salutation', 'Number of URLs', 'text link disparity',
       'Attachments', 'IP URLs', 'hexadecimal URL', 'Maximum Domains Counts',
       'Re: mail', 'contains account', 'contains prime targets'],
      dtype='object')
```

Figure 16: Univariate filtered features

iii) Feature importance with Random Forest Classifier

The Random Forest Classifier algorithm has built-in feature importance which I have computed with gini impurity. Following are the filtered features:

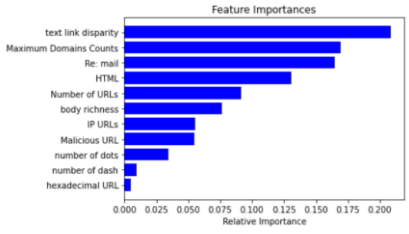


Figure 17: Features Importance with RandomForestClassifier

iv) High Correlation Filter

After filtering from above all filters, I union all the filtered features. Resultant is 15 features are:

```
{'Attachments',
 'General Salutation',
 'HTML',
 'IP URLs',
 'Malicious URL',
 'Maximum Domains Counts',
 'Number of URLs',
 'Re: mail',
 'body richness',
 'contains account',
 'contains prime targets',
 'hexadecimal URL',
 'number of dash',
 'number of dots',
 'text link disparity'}
```

Figure 18: Selected features

Now, I try apply high correlation filter on the resultant features. The heatmap is shown in figure 19.

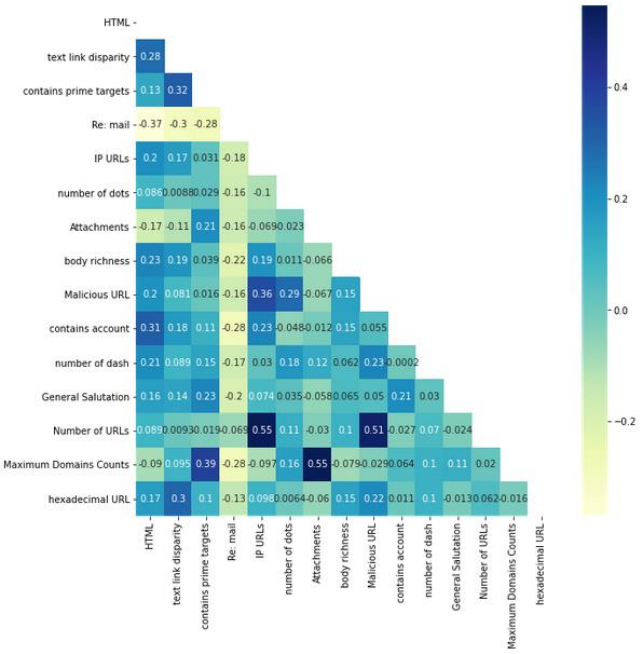


Figure 19: Selected features heatmap

This reveals that there is decent correlation between

a. Number of URLs and IP URLs
b. Maximum Domain Counts and Attachments
c. Number of URLs and Malicious URLs

We as until now I have 15 features and sacrificing 3 features would be not much beneficial, we will preserve this information. Finally,

we have all the 15 features selected for next phase i.e. model building.

4.4 Selecting best five model

In this phase, I have selected best five classifiers out of 11 classifiers as stated in previous section. I went ahead with using cross validation score to select best five classifiers which would be fine-tuned in later stage. As the dataset is slightly imbalanced I have selected f1 score as scoring for cross validation score. Following is the bar graph plotted in figure 20, where slight line on bar represent error rate.

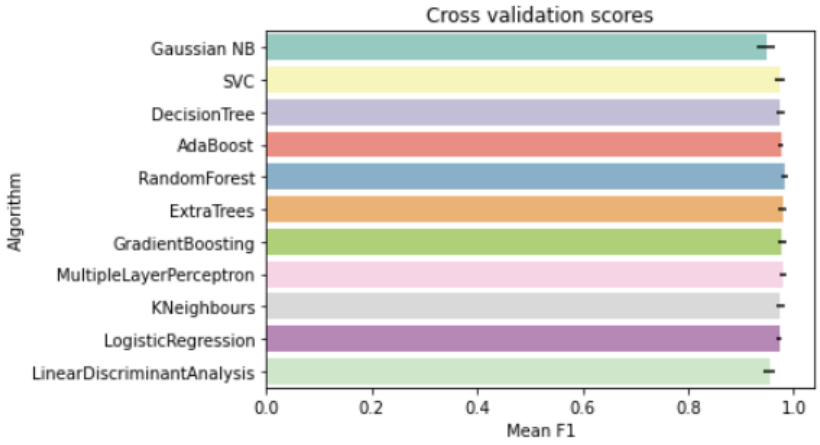


Figure 20: Cross Validation Scores (using f1 score) bar graph

CrossValMeans	CrossValerrors	Algorithm
0.898696	0.015845	Gaussian NB
0.958360	0.011144	LinearDiscriminantAnalysis
0.970543	0.006859	KNeighbours
0.971311	0.005626	DecisionTree
0.972172	0.004741	LogisticRegression
0.973549	0.007677	AdaBoost
0.977399	0.007659	SVC
0.978773	0.008664	MultipleLayerPerceptron
0.978873	0.009285	ExtraTrees
0.980328	0.007848	GradientBoosting
0.983509	0.007244	RandomForest

Table 6: Cross Validation Scores on 11 models

After observing mean f1 score and mean error, it is decided to select following five models for hyperparameter tuning.

i. SVC

ii. Logistic Regression

iii. Random Forest Classifier

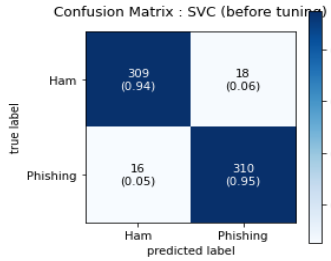
iv. ExtraTrees Classifier

v. Gradient Boosting

Following are the performances of models before hyperparameter tuning:

i. SVC

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.



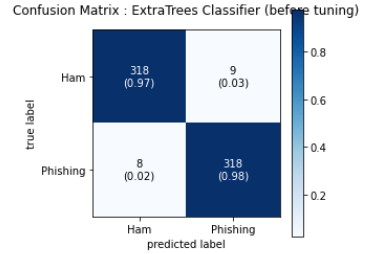
	precision	recall	f1-score	support
0	0.9508	0.9450	0.9479	327
1	0.9451	0.9509	0.9480	326
accuracy			0.9479	653
macro avg	0.9479	0.9479	0.9479	653
weighted avg	0.9479	0.9479	0.9479	653

MCC: 0.896
Log Loss: 0.139
f1: 94.801
Balanced Accuracy: 94.794
Accuracy: 94.793
AUC : 94.794

Figure 21: Performance of SVM after 10 fold cross validation

ii. ExtraTrees Classifier

This class implements a meta-estimator that fits a number of randomized decision trees (a.k.a. extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.



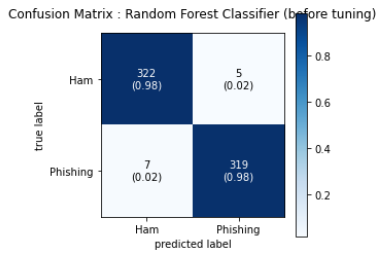
	precision	recall	f1-score	support
0	0.9755	0.9725	0.9740	327
1	0.9725	0.9755	0.9740	326
accuracy			0.9740	653
macro avg	0.9740	0.9740	0.9740	653
weighted avg	0.9740	0.9740	0.9740	653

MCC: 0.948
Log Loss: 0.101
f1: 97.397
Balanced Accuracy: 97.397
Accuracy: 97.397
AUC : 97.397

Figure 22: Performance of ExtraTrees Classifier after 10 fold cross validation

iii. Random Forest Classifier

A random forest [1] is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.



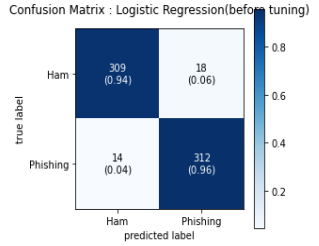
	precision	recall	f1-score	support
0	0.9787	0.9847	0.9817	327
1	0.9846	0.9785	0.9815	326
accuracy			0.9816	653
macro avg	0.9816	0.9816	0.9816	653
weighted avg	0.9816	0.9816	0.9816	653

MCC: 0.963
Log Loss: 0.14
f1: 98.154
Balanced Accuracy: 98.162
Accuracy: 98.162
AUC : 98.162

Figure 23: Performance of Random Forest Classifier after 10 fold CV

iv. Logistic Regression

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.



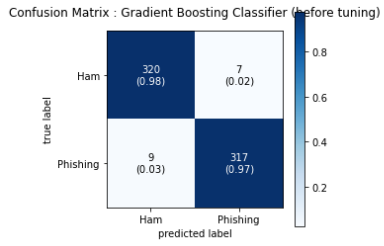
	precision	recall	f1-score	support
0	0.9567	0.9450	0.9508	327
1	0.9455	0.9571	0.9512	326
accuracy			0.9510	653
macro avg	0.9511	0.9510	0.9510	653
weighted avg	0.9511	0.9510	0.9510	653

MCC: 0.902
Log Loss: 0.205
f1: 95.122
Balanced Accuracy: 95.1
Accuracy: 95.1
AUC : 95.1

Figure 24: Performance of Random Forest Classifier after 10 fold CV

v. Gradient Boosting

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting.



	precision	recall	f1-score	support
0	0.9726	0.9786	0.9756	327
1	0.9784	0.9724	0.9754	326
accuracy			0.9755	653
macro avg	0.9755	0.9755	0.9755	653
weighted avg	0.9755	0.9755	0.9755	653

MCC: 0.951
Log Loss: 0.093
f1: 97.538
Balanced Accuracy: 97.549
Accuracy: 97.55
ROC_AUC : 97.549

Figure 25: Performance of Gradient Boosting Classifier after 10 fold CV

4.5 Hyperparameter tuning selected models

Hyperparameters tuning is crucial as they control the overall behavior of a machine learning model. A hyperparameter is a parameter whose value is set before the learning process begins. The ultimate goal is to find an optimal combination of hyperparameters that minimizes a predefined loss function to give better results.

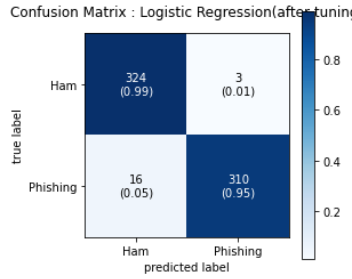
I have used Grid Search for hyperparameter tuning. This method tries every possible combination of each set of hyper-parameters. Using this method, we can find the best set of values in the parameter search space. This usually uses more computational power and takes a long time to run since this method needs to try every combination in the grid size.

Hyperparameters are crucial as they control the overall behavior of a machine learning model. The ultimate goal is to find an optimal combination of hyperparameters that minimizes a predefined loss function to give better results.

After hyperparameter, tuning performance of models along with their best parameters are:

i. Logistic Regression

```
LogisticRegression(C=2.5, solver='newton-cg')
```



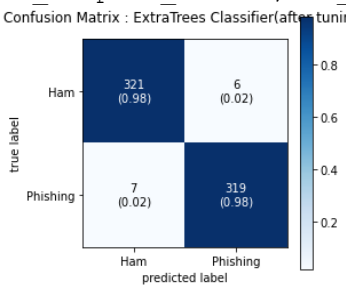
	precision	recall	f1-score	support
0	0.9529	0.9908	0.9715	327
1	0.9904	0.9509	0.9703	326
accuracy			0.9709	653
macro avg	0.9717	0.9709	0.9709	653
weighted avg	0.9716	0.9709	0.9709	653

```
MCC:  0.943
Log Loss:  0.162
f1:  97.027
Balanced Accuracy:  97.087
Accuracy:  97.09
ROC_AUC :  97.087
```

Figure 26: Performance of Logistic Regression after hyperparameter tuning

ii. Random Forest Classifier

```
RandomForestClassifier(bootstrap=False, max_features=3,
min_samples_leaf=3,min_samples_split=3)
```



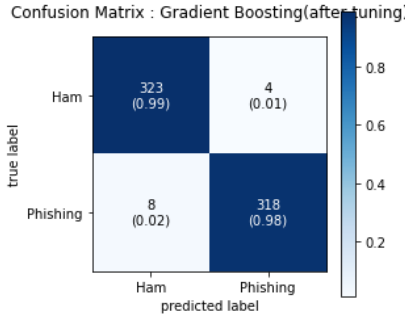
	precision	recall	f1-score	support
0	0.9787	0.9817	0.9802	327
1	0.9815	0.9785	0.9800	326
accuracy			0.9801	653
macro avg	0.9801	0.9801	0.9801	653
weighted avg	0.9801	0.9801	0.9801	653

```
MCC:  0.96
Log Loss:  0.067
f1:  98.003
Balanced Accuracy:  98.009
Accuracy:  98.009
AUC :  98.009
```

Figure 27: Performance of RF Classifier after hyperparameter tuning

iii. Gradient Boosting Classifier

```
GradientBoostingClassifier(max_depth=8, max_features=0.1,
min_samples_leaf=100,n_estimators=300)
```



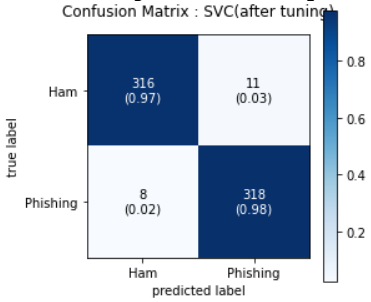
	precision	recall	f1-score	support
0	0.9758	0.9878	0.9818	327
1	0.9876	0.9755	0.9815	326
accuracy			0.9816	653
macro avg	0.9817	0.9816	0.9816	653
weighted avg	0.9817	0.9816	0.9816	653

```
MCC:  0.963
Log Loss:  0.072
f1:  98.148
Balanced Accuracy:  98.161
Accuracy:  98.162
AUC :  98.161
```

Figure 28: Performance of GB Classifier after hyperparameter tuning

iv. SVC

```
SVC(C=3, gamma=0.1, probability=True)
```



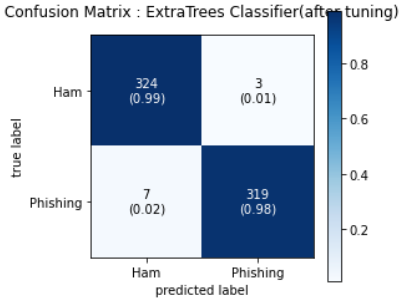
	precision	recall	f1-score	support
0	0.9753	0.9664	0.9708	327
1	0.9666	0.9755	0.9710	326
accuracy			0.9709	653
macro avg	0.9709	0.9709	0.9709	653
weighted avg	0.9709	0.9709	0.9709	653

```
MCC:  0.942
Log Loss:  0.11
f1:  97.099
Balanced Accuracy:  97.091
Accuracy:  97.09
AUC :  97.091
```

Figure 29: Performance of SVC after hyperparameter tuning

v. ExtraTrees Classifier

```
ExtraTreesClassifier(max_features=10, min_samples_split=10,
n_estimators=300)
```



	precision	recall	f1-score	support
0	0.9789	0.9908	0.9848	327
1	0.9907	0.9785	0.9846	326
accuracy			0.9847	653
macro avg	0.9848	0.9847	0.9847	653
weighted avg	0.9848	0.9847	0.9847	653

```
MCC:  0.969
Log Loss:  0.105
f1:  98.457
Balanced Accuracy:  98.468
Accuracy:  98.469
AUC :  98.468
```

Figure 30: Performance of ExtraTrees Classifier after hyperparameter tuning

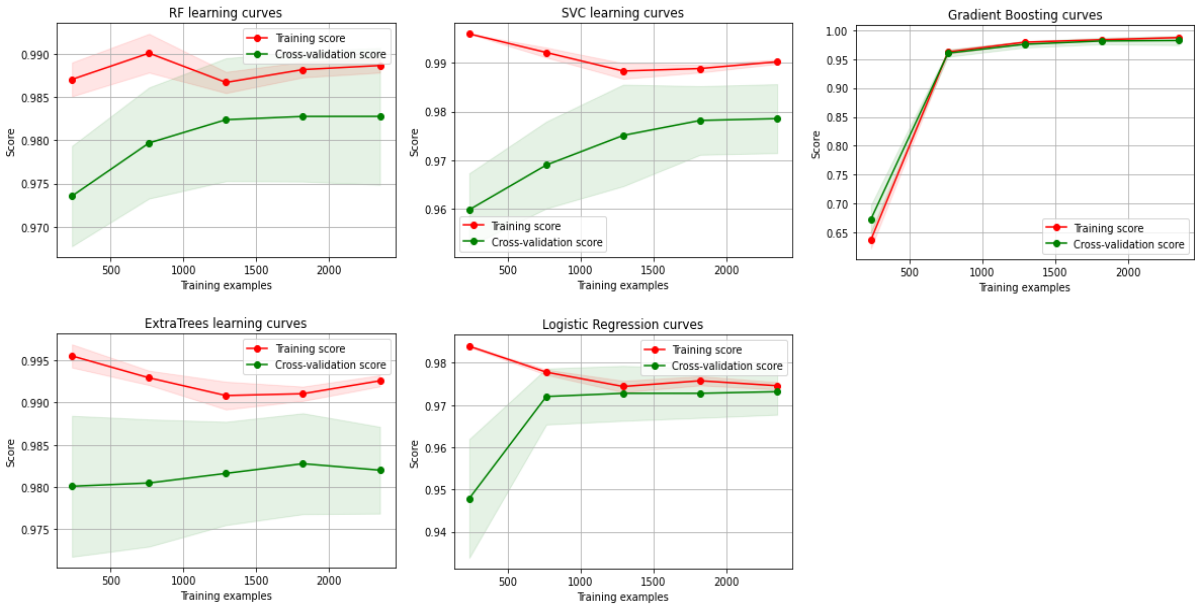


Figure 31: Learning Curve of tuned models

Now, if we try to focus a bit on learning curve it reveals that:

i. After a particular training size, SVC balances the bias and variance trade off and the curves are at almost same gap forward to the adequate training size.
ii. ExtraTrees also performed decent but after a point excess training leads to a slight increase in bias and then resulting underfitting.
iii. Random Forest also follows almost same trends as SVC, achieving decent balance between bias and variance.
iv. At last Logistic Regression and GradientBoosting are highly overfitting.

So, for voting classifier we will go with three models:

a. SVC (C=3, gamma=0.1, probability=True)
b. ExtraTrees Classifier (max_features=1, min_samples_split=10, n_estimators=300)
c. Random Forest Classifier (bootstrap=False, max_features=3, min_samples_leaf=3,min_samples_split=3)

Above all three have good balanced accuracy and also do well on balancing the bias and variance trade off.

4.6 Voting Ensemble Classifier on top of selected models

A Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output (class) based on their highest probability of chosen class as the output.

It simply aggregates the findings of each classifier passed into Voting Classifier and predicts the output class based on the highest majority of voting. The idea is instead of creating separate dedicated models and finding the accuracy for each them, we create a single model which trains by these models and predicts output based on their combined majority of voting for each output class.

A voting ensemble works by combining the predictions from multiple models. It can be used for classification or regression. In the case of classification, the predictions for each label are summed and the label with the majority vote is predicted.

Voting Classifier supports two types of voting.

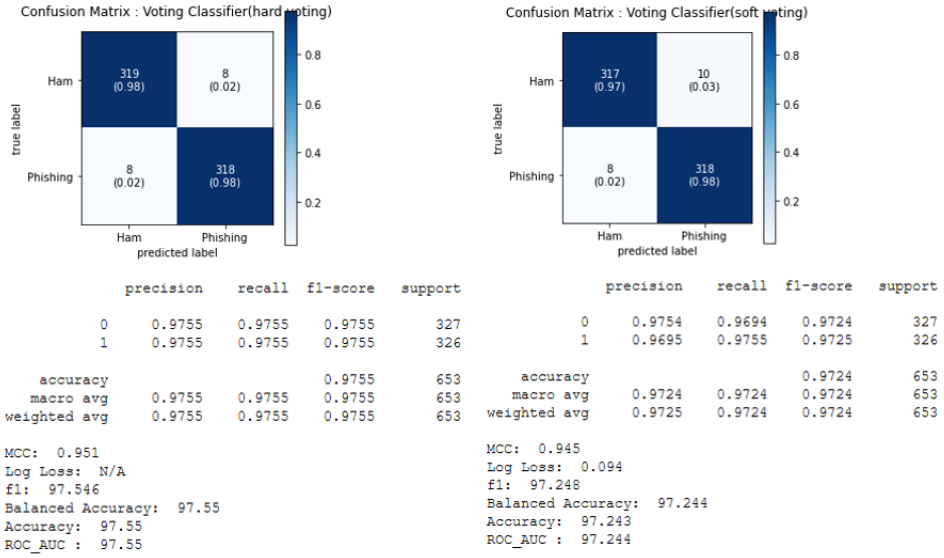
Hard Voting: In hard voting, the predicted output class is a class with the highest majority of votes i.e. the class which had the highest probability of being predicted by each of the classifiers. Suppose three classifiers predicted the output class(A, A, B), so here the majority predicted A as output. Hence A will be the final prediction.

Soft Voting: In soft voting, the output class is the prediction based on the average of probability given to that class. Suppose given some input to three models, the prediction probability for class A = (0.30, 0.47, 0.53) and B = (0.20, 0.32, 0.40). So the average for class A is 0.4333 and B is 0.3067, the winner is clearly class A because it had the highest probability averaged by each classifier

In our case, we took 3 classifiers after hyperparameter tuning under the Voting Classifier. They are:

a. SVC (C=3, gamma=0.1, probability=True)
b. ExtraTrees Classifier (max_features=1, min_samples_split=10, n_estimators=300)
c. Random Forest Classifier (bootstrap=False, max_features=3, min_samples_leaf=3,min_samples_split=3)

We performed both soft and hard voting and results are as follows:



	precision	recall	f1-score	support
0	0.9755	0.9755	0.9755	327
1	0.9755	0.9755	0.9755	326
accuracy			0.9755	653
macro avg	0.9755	0.9755	0.9755	653
weighted avg	0.9755	0.9755	0.9755	653

MCC: 0.951
Log Loss: N/A
f1: 97.546
Balanced Accuracy: 97.55
Accuracy: 97.55
ROC_AUC : 97.55

	precision	recall	f1-score	support
0	0.9754	0.9694	0.9724	327
1	0.9695	0.9755	0.9725	326
accuracy			0.9724	653
macro avg	0.9724	0.9724	0.9724	653
weighted avg	0.9725	0.9724	0.9724	653

MCC: 0.945
Log Loss: 0.094
f1: 97.248
Balanced Accuracy: 97.244
Accuracy: 97.243
ROC_AUC : 97.244

Figure 32: Performance of Voting Classifiers on test data

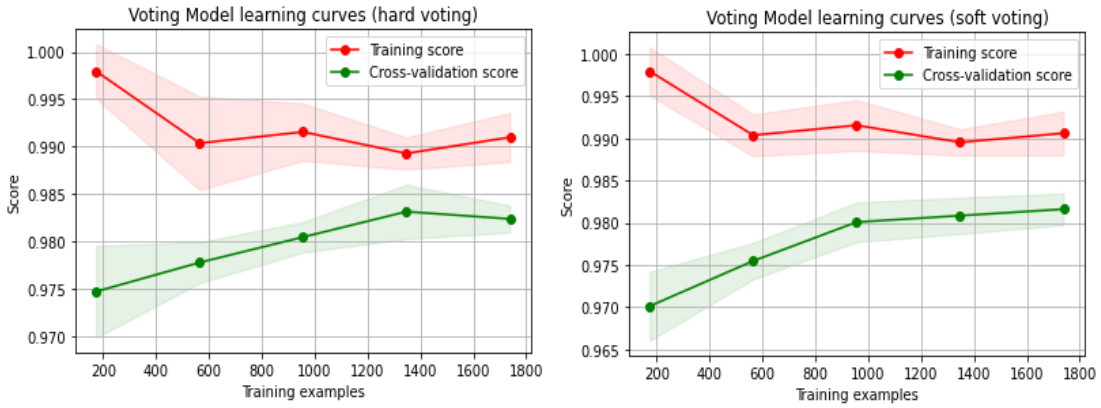


Figure 33: Learning curve of Voting Models

Both voting methods performed almost exactly same. Athough they have slight less accurate, yet they have very good balance in bias and variance trade off. They can be very useful in real world application as they tend not to overfit the training data while minimizing the gap between training and cross validation error. Thus, it could be inferred that they will perform better in real world application.

4.7 Performance comparison

As our models are trained, cross validated and tested, it's time to compare them. As our dataset was slightly imbalanced, we should avoid concentrating on accuracy rather focus on other parameters as balanced accuracy, f1 score, MCC etc. They would better represent the performance of the models.

Table 7 shows the comparative performance of all the models considering various performance metrics in consideration. Figure 34 shows the bar plot of different models performance on the basis of balanced accuracy (it is used as our dataset is slightly imbalanced).

Model	Accuracy	Balanced Accuracy	Log loss	F1 Score	MCC	ROC_AUC
Logistic Regression(before tuning)	95.100	95.100	0.205	95.122	0.902	95.100
Logistic Regression(after tuning)	97.090	97.087	0.162	97.027	0.943	97.087
SVC (before tuning)	94.793	94.794	0.139	94.801	0.896	94.794
SVC(after tuning)	97.090	97.091	0.11	97.099	0.942	97.091
Gradient Boosting Classifier (before tuning)	97.550	97.549	0.093	97.538	0.951	97.549
Gradient Boosting(after tuning)	98.009	98.008	0.072	97.991	0.960	98.008
ExtraTrees Classifier (before tuning)	97.550	97.550	0.092	97.546	0.951	97.550
ExtraTrees Classifier(after tuning)	98.469	98.468	0.071	98.462	0.969	98.468
Random Forest Classifier (before tuning)	98.315	98.315	0.143	98.310	0.966	98.315
Random Forest(after tuning)	98.469	98.468	0.066	98.457	0.969	98.468
Voting Classifier(hard voting)	97.550	97.550	N/A	97.546	0.951	97.550
Voting Classifier(soft voting)	97.243	97.244	0.094	97.248	0.945	97.244

Table 7: Performance comparison of various models

Although all the models performed well after tuning up the hyperparameters, we noticed that the Voting Classifier didn't outperform

the estimators it pools votes from. The voting classifier perform mostly similar in both soft and hard voting and nicely balances the bias and variance tradeoff.

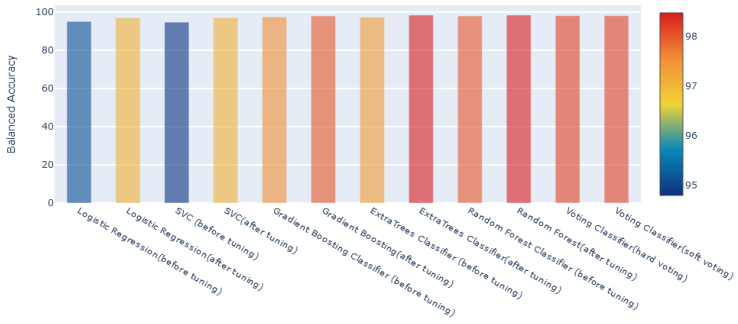


Figure 34: Bar plot of performance of different models

During the project both ExtraTrees Classifier and Random Forest were really close to each other performance wise. Although balanced accuracy of both the ExtraTrees Classifier and Random Forest are same yet Random Forest has good log loss score in comparison to ExtraTrees.

Random Forest came out as the better model to identify the phishing mail considering the selected features than ExtraTrees as it has decent balance of bias and variance. SVC would also be a really good choice in real world as shown in project it smoothly balances the bias and variance tradeoff thus it learns while training still it doesn't overfit and can perform really good in new data.

The bar plot above shows how the models performed in terms of balanced accuracy score. We can see every model has improved the performance after tuning up.

Following inferences are made from the project :

A. Best performing model is hyperparameter tuned Random Forest Classifier with accuracy of 98.469% and log loss of 0.066.and parameters are (bootstrap=False,max_features=3,min_leaf_sample=3,min_sample_split=3)

B. Although Random Forest performed best in our project experiment, SVM and Voting Classifier turns out to be better in balancing bias and variance tradeoff.

C. Logistic Regression and Gradient Boosting tend to overfit so better to avoid or use in a way that can counter the overfitting. For example, we can include them in model stacking or voting classifier and counter the overfit by adjusting the weights.

D. Lastly, **it is evident that to have better real world performing model we can go for Voting Classifier containing above described hyperparameter tuned classifiers to have better performance to newer test data**.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this project, the accuracy of phishing email detection were evaluated based on manually determining which feature to extract from mail and automated feature selection through various filters. Finally, comparison among the performance of models is done.

The project achieved accuracy of 98.469% by using the Random Forest Classifier and ExtraTrees both. The log loss of the Random Forest is much less than the Extra Trees and also has a better learning curve. So Random Forest emerges as best performing model among all the models. The real interesting thing revealed is even ExtraTrees Classifier has accuracy similar to Random Forest, it underfits if training size increases and Random Forest may overfit if training size is less or moderate.

SVC having the accuracy of 97.09% had came out as best in balancing the variance and bias tradeoff. It smoothly learns while training but it does not overfit or underfit. SVC could be best model for real world application.

Finally, the voting classifier had also performed quite good achieving 97.553% accuracy but it has very good bias and variance tradeoff which makes is best choice for real world application.

5.2 Future Work

Feature selection techniques need more improvement to cope with the continuous development of new techniques by the phishers over the time. Therefore, it is recommended to developing a new automated tool in order to extract new features from new raw emails to improve the accuracy of detecting phishing email and to cope with the expanding with phisher techniques.

References

[1] Andronicus A. Akinyelu and Aderemi O. Adewumi, "Classification of Phishing Email Using Random Forest Machine Learning Technique", Hindawi Publishing Corporation, Journal of Applied Mathematics, Volume 2014, Article ID 425731

[2] Patrick Lawson, Carl J. Pearson, Aaron Crowson, Christopher B. Mayhorn, "Email phishing and signal detection: How persuasion principles and personality influence response patterns and accuracy", Applied Ergonomics 86 (2020) 103084

[3] Adwan Yasin and Abdelmunem Abuhasan, “An intelligent classification model for Phishing email detection”, International Journal of Network Security & Its Applications (IJNSA) Vol.8, No.4, July 2016

[4] Jingguo Wang, Tejaswini Herath, Rui Chen, Arun Vishwanath, And H. Raghav Rao, “Phishing Susceptibility: An Investigation Into the Processing of a Targeted Spear Phishing Email”, IEEE transactions on professional communication, vol. 55, no. 4, December 2012

[5] Tushaar Gangavarapu, C. D. Jaidhar · Bhabesh Chanduka, “Applicability of machine learning in spam and phishing email filtering: review and approaches”, Artificial Intelligence Review volume 53, pages 5019–5081 (2020)

[6] Meenakshi Das, Sowmya Saraswathi, Rashmi Panda, Alekha Kumar Mishra and Asis Kumar Tripathy, “Exquisite Analysis of Popular Machine Learning–Based Phishing Detection Techniques for Cyber Systems”, Journal of Applied Security Research, September 2020, ISSN: 1936-1629, DOI: 10.1080/19361610.2020.1816440

[7] Alekha Kumar Mishra, Asis Kumar Tripathy, Sowmya Saraswathi, and Meenakshi Das, “Prevention of Phishing Attack in Internet-of-Things based Cyber-Physical Human System”, High Performance Vision Intelligence, Springer, 2020, EISBN: 978-981-15-6844-2, DOI:10.1007/978-981-15-6844-2

[8] Alkhalil Z, Hewage C, Nawaf L and Khan I (2021), “Phishing Attacks: A Recent Comprehensive Study and a New Anatomy”, Front. Comput. Sci. 3:563060. doi: 10.3389/fcomp.2021.563060

[9] Schuetzler, Ryan M., "Trends in Phishing Attacks: “Suggestions for Future Research”, Information Systems and Quantitative Analysis Faculty Proceedings & Presentations. 25.

[10] El Aassal, A., Baki, S., Das, A., & Verma, R. M. (2020). “An in-depth benchmarking and evaluation of phishing detection research for security needs” .IEEE Access, 8, 22170–22192

[11] R. Suriya, K. Saravanan, Arunkumar Thangavelu, “An integrated approach to detect phishing mail attacks: a case study”, Association of Computing Machinery, Proceedings of the 2nd international conference on Security of information and networks, October 2009 Pages 193–199 https://doi.org/10.1145/1626195.1626244

[12] Jayshree Hajgude, Lata Ragha, “Phish mail guard: Phishing mail detection technique by using textual and URL analysis”, IEEE, 2012 World Congress on Information and Communication Technologies, https://doi.org/10.1109/WICT.2012.6409092

[13] Marforio, C., Masti, R. J., Soriente, C., Kostiainen, K., and Capkun, S. (2015). “Personalized security indicators to detect application phishing attacks in mobile platforms”. Available at: http://arxiv.org/abs/1502.06824.

[14] Ponnurangam Kumaraguru, Yong Rhee, Alessandro Acquisti, Lorrie Faith Cranor, Jason Hong, Elizabeth Nunge, "Protecting people from phishing: the design and evaluation of an embedded training email system", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, April 2007 Pages 905–914, https://doi.org/10.1145/1240624.1240760

[15] Parmar, B. (2012). Protecting against spear-phishing. Computer Fraud & Security, 8-11. Ramanathan, V., & Wechsler, H. (2012). phishGILLNET—phishing detection methodology using probabilistic latent semantic analysis, AdaBoost, and co-training. EURASIP Journal on Information Security, 1-22.

[16] Kumaraguru, P., Sheng, S., Acquisti, A., Cranor, L. F., & Hong, J. (2010). Teaching Johnny not to fall for phish. ACM Transactions on Internet Technology (TOIT), 10(2), 7.

[17] Cao, Y., Han, W., & Le, Y. (2008). Anti-phishing based on automated individual white-list. In Proceedings of the 4th ACM workshop on Digital identity management (pp. 51-60).

[18] Chandrasekaran, M., Narayanan, K., & Upadhyaya, S. (2006, June). Phishing email detection based on structural properties. In NYS Cyber Security Conference (pp. 1-

[19] Adida, B., Chau, D., Hohenberger, S., & Rivest, R. L. (2006). Lightweight email signatures. In Security and Cryptography for Networks (pp. 288-302). Springer Berlin Heidelberg.

[20] Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007, October). A comparison of machine learning techniques for phishing detection. InProceedings of the anti-phishing working groups 2nd annual eCrime researchers summit (pp. 60-69). ACM.

[21] Steinberg, D., & Colla, P. (2009). CART: classification and regression trees. The top ten algorithms in data mining, 9, 179.

[22] Bergholz, A., Chang, J. H., Paass, G., Reichartz, F., & Strobel, S. (2008, August). Improved Phishing Detection using Model-Based Features. InCEAS.

[23] Toolan, F., & Carthy, J. (2009, September). Phishing detection using classifier ensembles. In eCrime Researchers Summit, 2009. eCRIME'09.(pp. 1-9). IEEE.

[24] Gansterer, W. N., & Pölz, D. (2009). E-mail classification for phishing defense. In Advances in Information Retrieval (pp. 449-460). Springer Berlin Heidelberg.

[25] Ma, L., Yearwood, J., & Watters, P. (2009, September). Establishing phishing provenance using orthographic features. In eCrime Researchers Summit, 2009. eCRIME'09. (pp. 1-10). IEEE.

[26] Basnet, R. B., & Sung, A. H. (2010). Classifying phishing emails using confidence-weighted linear classifiers. In International Conference on Information Security and Artificial Intelligence (ISAI) (pp. 108-112).

[27] Wu, Y., Zhao, Z., Qiu, Y., & Bao, F. (2010, May). Blocking foxy phishing emails with historical information. In Communications (ICC), 2010 IEEE International Conference on (pp. 1-5). IEEE.

[28] Khonji, M., Iraqi, Y., & Jones, A. (2013). Enhancing phishing E-Mail classifiers: a lexical URL analysis approach. International Journal for Information Security Research (IJISR), 2(1/2).

[29] Alguliev, R. M., Aliguliyev, R. M., & Nazirova, S. A. (2011). Classification of textual e-mail spam using data mining techniques. Applied Computational Intelligence and Soft Computing.

[30] Al-Momani, A., Wang, T. C., Altaher, A., Manasrah, A., Al-Momani, E., Anbar, M., Ramadass, S. (2012). Evolving fuzzy neural network for phishing emails detection, Journal of Computer Science, 8, 1099.

[31] Kumar, R. K., Poonkuzhali, G., & Sudhakar, P. (2012, March). Comparative study on email spam classifier using data mining techniques. In Proceedings of the International MultiConference of Engineers and Computer Scientist (Vol. 1, pp. 14-16).

[32] Jameel, Noor Ghazi M., and Loay E. George. Detection of phishing emails using feed forward neural network. International Journal of Computer Applications 77 2013.

[33] Zhang, N., & Yuan, Y. (2013). Phishing detection using neural network. Department of Computer Science, Department of Statistics, Stanford University. Web.

[34] Rathi, M., & Pareek, V. (2013). Spam Mail Detection through Data Mining-A Comparative Performance Analysis. International Journal of Modern Education and Computer Science, (12), 31.

[35] Akinyelu, A. A., & Adewumi, A. O. (2014). Classification of phishing email using random forest machine learning technique. Journal of Applied Mathematics.

[36] Nizamani, S., Memon, N., Glasdam, M., & Nguyen, D. D. (2014). Detection of fraudulent emails by employing advanced feature abundance. Egyptian Informatics Journal, 15(3), 169-174.

[37] Kathirvalavakumar, T., Kavitha, K., & Palaniappan, R. (2015). Efficient Harmful Email Identification Using Neural Network, British Journal of Mathematics & Computer Science.

[38] https://www.frontiersin.org/articles/10.3389/fcomp.2021.563060/full#B72, Figure 5, The growth in phishing attacks 2015–2020 by quarters based on data collected from APWG annual reports.