KENYATTA UNIVERSITY

A MACHINE LEARNING MODEL TO DETECT PHISHING EMAILS USING ENSEMBLE TECHNIQUE.

Fredrick Nthurima Murangiri (BSc).

J57/38638/2016

Signature_____ Date_____

Department of Computing & Information Science

"A RESEARCH PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE IN THE SCHOOL OF PURE AND APPLIED SCIENCES OF KENYATTA UNIVERSITY"

Dr. Abraham Mutua, Dr. Stephen Waithaka,

Signature: _____ Date: _____Signature: _____Date: _____

Department of Computing & Information Science Department of Computing & Information Science

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

EMBER - Endgame Malware Benchmark for Research

PIN - Personal Identification Number

IBM – International Business Machines

URL - Uniform Resource Locator

DEFINITION OF TERMS

Classification is dividing data into groups or categories based on characteristics.

Algorithm - a set of instructions for solving a problem or accomplishing a task

Cyber security – refers to defending of internet devices like computers, servers, mobile phones, electronic systems, networks, and data from spiteful attacks.

Machine learning - the advancement of PC frameworks that can adjust to guidelines naturally utilizing calculations and measurable models to dissect and draw derivations from information designs.

Spam Emails – unasked-for email messages, often sent in bulk to a vast range of recipients

Cyberattack – demonstration of acquiring unapproved admittance to a PC, figuring framework or PC network with the reason to cause harm.

Intrusion Detection - monitoring network traffic for suspicious activity and alert when such activity is discovered.

Phishing emails – the process stealing sensitive information by sending an email that falsely appears to be from a legitimate organization.

Random Forest - is a gathering learning technique for grouping, relapse, and different assignments that works by developing numerous decision trees at preparation time.

ABSTRACT

The majority of phishing attacks prey on behavioral flaws in users. Phishing links are included in an email that an attacker sends to the recipient that looks and feels authentic. The attacker can obtain sensitive data, like as usernames, passwords, and credit card details, by having the receiver click on the embedded links and access the hacked account. With the increasing case of cyber-attacks, organizations are looking for safer ways of protecting data and preventing getting hacked or getting hacked again. Design and technology should be greatly improved to prevent hackers from infiltrating networks. Phishing attacks, which mostly target financial organizations, have been identified as the most common online content attack according to surveys. A 2017 Ponemon Institute LLC survey estimated that the yearly loss from phishing attempts is almost $1.5 billion. The Internet of Things (IoT) is contributing to the global danger to information security; hence, a more effective phishing detection system is needed to reduce these losses and reputation injury. In order to increase the accuracy of phishing detection and prevention, this research study investigates and reports the use of several machine-learning models by utilizing more phishing email features and the random forest algorithm. To detect and prevent phishing attacks, this project examined current phishing techniques, examined the impact of using an ensemble model, designed and created a supervised classifier to identify and stop phishing emails, and tested the model using available data. The model was learned under supervision using a dataset of legitimate and fraudulent emails. With a rate of less than 0.1% for False Positives (FP) and False Negatives (FN), the expected accuracy is 99.9% which will be higher than the already existing models therefore better detection of fraudulent emails.

CHAPTER ONE

INTRODUCTION

1.1 Introduction

This chapter presents the background of the study, statement of the problem, research objectives, research questions, the justification and the scope of the study. Finally, the assumptions made during the research work are presented. The majority of phishing attacks prey on behavioral flaws in users. An attacker sends the receiver an email that appears to be authentic but contains links to a phishing website. The attacker can obtain sensitive data, like as usernames, passwords, and credit card details, by having the receiver click on the embedded links and access the hacked account.

Based on the Anti-Phishing Working Group Report 2022, a Phishing attack is the number one attack committed by threat actors as compared to other attacks. It is a form of fraud where the attacker deceives the target for personal gain or reputation damage. Fraud results in users revealing their details like credit card numbers, passwords, PINs, usernames and other sensitive information leading to the compromise of accounts and loss of funds.

Phishing campaigns lure users into giving confidential information by visiting websites that look like legitimate ones, phishing.org, (2018). Phishing is done using a digital gadget like a computer or Ipad through a computer network. Malicious actors usually target the weakest element in the security chain, i.e., end-users Khonji.M, Jones. A and Iraqi. Y, (2013).

With a phishing attack, the attackers package messages so the target users cannot easily detect if the message is not genuine. The users end up clicking on the embedded links, thereby being redirected to the attacker's websites, whereby the attacker can get confidential information like passwords, usernames, credit card numbers etc. This enables threat actors to enter the compromised account and achieve their objectives like data theft, funds transfer or reputation injury.

For instance, a malicious email might have malware which, when clicked by the user, will install itself in the pc or mobile phone and will transfer funds to the account of the attacker whenever the owner of the account tries to transfer cash ,Khonji. M et al., (2013). This attack is called Man in the Browser (MITB), a variant of the Man in the Middle (MITM) attack. The man-in-the-browser attack usually uses vectors like ActiveX components, plugins, or email attachments to deliver the payload to the user's computer or phone.

With the increasing case of cyber-attacks, organizations are looking for safer ways of protecting data and preventing getting hacked or getting hacked again. Design and technology should be greatly improved to prevent hackers from infiltrating networks.

According to Behdad.M, French.T, Bennamoun.M, and Barone. L, (2012), using better defense systems is not enough to stop malicious actors from penetrating systems since these are sometimes circumvented; a better system should detect malicious activities and prevent them before causing any damage.

Several mechanisms are used today to filter spam but are static, so they cannot handle the ever-evolving threats and phishing trends. They can only detect already known phishing patterns leaving behind future attacks. This is a security weakness because attackers are not static and use different ways of evading detection. This challenge has motivated researchers to look for other ways to detect known and new threats. As a result, machine learning algorithms have come into the knowledge of many researchers and information security professionals to curb phishing attacks.

Machine learning (ML) is a discipline of artificial intelligence that utilizes mining of data to detect fresh and old phishing features retrieved from a given information set which is ultimately utilized for the classification of benign and phishing emails.

In this project, multiple models were used using an ML algorithm, namely random forest. Moreover, phishing features are known from the literature, and a combination of 15 (fifteen) features were be used. We used a dataset consisting of 6000 emails. These emails comprised of legitimate and phishing emails from where the phishing features were extracted. A vector representation was formed with the extracted features, which was used to train our classifier model.

1.2 Statement of the Problem

Today, many spam email filters exist compared to filters for phishing emails. Many techniques are employed to develop phishing email filters, including Blacklists, Visual similarity, heuristics, and Machine Learning.

The results of the above techniques have shown that Machine Learning do offer the best solution for phishing filters, Brown. et al., (2017).

However, current machine-learning anti-phishing solutions use a single model to detect phishing. According to the results, this could offer better detection accuracy, which currently stands at 98% Smadi et al.,(2015). Moreover, they have used domain/URL characteristics, leaving behind other phishing features in phishing emails and lowering accuracy and detection rates.

There was a need to develop a better phishing classifier using a machine learning ensemble model, namely Random Forest, and include other phishing email features to increase detection accuracy. The random forest algorithm (RF) employed in this project work is a form of a bagging algorithm that categorizes many decision trees from random training sets) to get improved classification accuracies, Deng et al., (2020).

1.3 Justification

This research create users' awareness of how attackers steal credentials to infiltrate systems and how to prevent the attack. The research developed a better email classification model with higher accuracy.

1.4 Research questions

1. How do attackers lure users to visit phishing websites?
2. Can the use of an ensemble model and more features lead to increased accuracy in phishing detection?

3. To what accuracy can this ensemble model achieve phishing detection?
4. What recommendations can be inferred for future ***classifiers***?

1.5 Hypothesis

Use of ensemble model and more email features leads to increased phishing email detection accuracy.

1.6 Objectives

1.6.1 Purpose of the Study

This research aimed to develop a classifier model to detect phishing emails with an improved accuracy rate and reduced mean square error.

1.6.2 Specific Objectives

The project was directed at achieving the objectives listed below;

1. To investigate the existing phishing attack methods used by attackers to lure users.
2. To investigate the effect of using an ensemble model and more features in phishing email detection.
3. To design and develop a supervised classifier model which can detect phishing emails
4. To test the classifier model with existing data

1.7 Significance of the study

Phishing is a global problem affecting all sectors of the economy, ranging from banking, manufacturing, health and personal privacy. When personal information and credentials fall into the wrong hands, there is a risk of losing funds or reputation damage. The finding

of this research could assist organizations and individuals in strengthening their security posture and prevent them from becoming phishing victims.

1.8 Scope

This research covers phishing attacks, how they are propagated and prevention. The project developed a classifier model to make better the accuracy of the existing models. The model used an ensemble technique with fifteen learning features. The project did not cover phishing attacks originating from non-electronic means.

1.9 Assumptions

The phishing attack is delivered in digital form.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter explores phishing attack vectors, statistics in phishing attacks, how the phishing act works, and how different phishing attack filters work. The majority of phishing attacks prey on behavioral flaws in users. An attacker impersonating a sender and including phishing links in an email that appears to be authentic. The attacker can obtain sensitive data, like as usernames, passwords, and credit card details, by having the receiver click on the embedded links and access the hacked account.

2.2 Understanding Phishing Attacks

As per the Anti-Phishing Working Gathering (APWG) report, the name "Phishing Movement Patterns Report - fourth Quarter 2017", around 57% of phishing assaults target monetary foundations and settlement administrations. A phishing attack is a very common threat on the internet propagated by malicious actors who lure users into supplying personal information to their websites. By doing so, the malicious actors are able to harvest critical information about a user ranging from passwords, credit card numbers or usernames, for their malicious objectives.

Researchers have demonstrated that social phishing, where in this case, the word *social* means information related to the target is used, produces very actual results as opposed to regular phishing. Gupta.M, Prakash. P, Kompella. R, and Kumar.M, (2015) concluded that if phishing attack emails mimicked a target’s ally, the success rate of the phishing attack

grew from 16% to 72%. Information's social aspect is valuable to social network operators and attackers. This is made even more possible if the information on social media contains an email address that is genuine or if there is a recent conversation between the target and the mimicked friend.

In the recent past, there has been an emergence with automation of data extraction from social media networks and sites. This has led to the availability of usable data to attackers, which can be used to carry out phishing attacks.

Ofoghi. B, Ma. L, Watters. P and Brown.S, (2017) grouped the following spam attacks; Shared attribute attacks, Relationship-based attacks and Unshared attribute attacks.

With this kind of grouping, Relationship-based attacks use affiliation information only, making this spam attack look like socially engineered phishing which normally tricks users into clicking and inputting sensitive data. With the other attacks, they use information originating from social networks to compromise users and get sensitive data for their malicious actions.

This information originating from social networks is categorized between shared and unshared concerning the target and spoofed friend. Birthday cards can represent unshared information, which looks like genuine cards sent from the target's friend. On the other hand, common attributes like photos where both the victim and mimicked friends are both tagged can be abused for context-aware spam.

Huber M, Mulazzani M, Leithner M, Schrittwieser S, Wondracek G, Weippl E ,(2011) found that information from various social networks can be abused as a result of weaknesses in the communication channels. This enables the attackers to acquire sensitive information for their gain. Furthermore, the authors have gone further to demonstrate that the data that is extracted from online networks can be exploited to aim many users with context-ware spam.

Gupta.M, Prakash. P, Kompella. R and Kumar.M, (2015) used a hybrid of two techniques, namely blacklists and heuristics, to detect phishing emails. This hybrid technique attained a False Positive (FP) of 5% and a False Negative (FN) rate of 3%.

Holbrook, M., Kumaraguru, P., Downs, J., Cranor, L.F. and Sheng, S.,(2010) researched several anti-phishing solutions and came up with '*SpoofGuard,*' which was designed by Ledesma. R, Chou. N, Mitchell. C and Teraguchi, Y, (2014). This solution '*SpoofGuard*' showed an improved detection rate of 38% for False Positives (FP) and 9% for False negatives (FN). Moreover, S. Nargundkar, N. Tiruthani, and W. D. Yu (2017) developed a phishing detection system that used heuristics as a mode of detection. This solution managed to achieve a False positive of 1% and a False Negative of 20%.

Smadi, S., Aslam, N., Zhang, L., Alasem, R. & Hossain, M.A. (2015) also used the heuristics technique, which achieved a False Positive rate of 3% and 11% False Negative.

Sadeh,N, Fette, & Tomasic,A.,(2017) Came up with a solution to detect phishing emails by use of Machine Learning. This technique achieved a False Positive rate of 1% and a False Negative rate of 1.2%. Strobel,S, Glahn,S., Moens,M.F., De Beer,J., and

Bergholz,A., (2010) Came up with a hybrid solution by use of machine learning and heuristics, which achieved a False Positive rate of 0.05% and a False Negative rate of 1%. The above techniques have relatively high False Positives and False Negatives. In our proposed anti-phishing technique, the features are extract5434ed directly from the email, thus eliminating processing overhead and increasing run-time. Thus, by eliminating sending of queries, the model was faster and removed space complexities.

2.3 Machine Learning Anti-Phishing Methods.

2.3.1 PhishHaven—An Efficient Real-Time AI Phishing URLs Detection System

This constant computer-based intelligence-produced phishing URL arrangement of recognition was created by Maria Sameen, Kyunghyun Han and Seong Oun Hwang in 2020. This framework utilizes lexical elements-based extraction and investigation techniques. To expand the productivity of the framework, the framework utilizes URL HTML encoding as a lexical element. To detect tiny URLs, the system uses a URL hit mechanism. This system uses an ensemble machine learning model employing the multi-threading approach for the training and testing stages.

The framework utilizes fair democracy to allocate the last marks, i.e., typical or phishing, to the given URLs. This framework accomplished an exactness pace of 98% discovery. The framework involves a worldview execution for troupe AI, which includes equal execution of learning models through multi-stringing. Equal execution in the preparing and testing stages speeds up processes, consequently permitting the location of phishing URLs to progress. The proposed recognition framework flaunts different helpful highlights. First, it is free of any outsider administrations (i.e., WHOIS, Group Cymru, and so on) because

every one of the methods, including highlight extraction from a URL assessment and characterization of a URL, is performed inside our location framework. Second, it is free of dialects since it dissects URLs, as it were. Furthermore, third, it can recognize zero-day assaults because the discovery framework dissects URLs in view of the URL's lexical highlights.

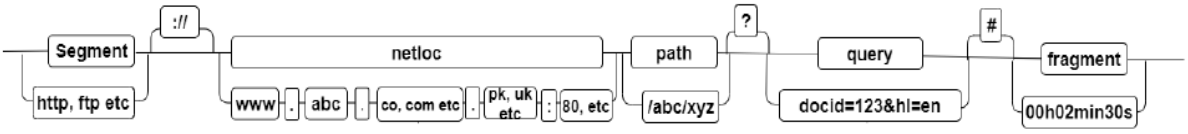


Figure 2.1: Lexical features approach from components of URL

2.4 Current Application of Phishing Classifiers.

Phishing emails exhibit different features that make them get distinguished from benign emails. These features include; subdomain, prefix_suffix, URL length etc. Mohammad, R. M., Thabtah, F. & McCluskey, L. (2013) created unique learning bases using space understanding to detect phishing and legitimate emails. Recent research shows on how to automate the detection of benign and phishing emails. The use of statistical analysis has been used to achieve this, according to Abdelhamid N., Thabtah F., and Ayesh A., (2014). To study phishing emails better, emails from various sources were grouped based on various phishing features. This grouping was achieved through the recording of occurrences of phishing emails. To improve the detection rate, a larger dataset was collected from various sources Abdelhamid N., Thabtah F., Ayesh A., (2014).

Several methods have been used to study phishing patterns. These methods are decision tree, support vector machine, random forest and Naïve Bayes. A solution called PILFER, which stands for "*Phishing Identification by Learning on Features of Email Received*," was

designed to help curb the phishing menace. This solution was used with a case study of 860 phishing emails and 695 benign emails. This experiment was conducted to determine the phishing features in the emails. The features detected by this solution and experiment include IP-based URLs, Email body in HTML format, presence of javascripts, number of links inside the email and others. Therefore, it was found that PILFER is good at improving the detection of phishing emails by considering the features founds in the emails.

A method called the Random Forest algorithm was used against 2000 email messages. This experiment aimed to reduce false positives and false negative rates Akinyelu A. A. and Adewumi A. O.,(2014). When Random Forest is used with a combination of 15 features, it registers a significant reduction in error rate, becoming the best method in phishing classification and detection hence fitting. Phishing detection models using Random Forest are more dominant concerning detection rate.

Aburrous M., Hossain M., Dahal KP and Thabtah F. (2010) used identified features to classify websites by accurately classifying the identified features. The manual classification was used to group these features into six categories. The categories were then loaded into Waikato Environment for Knowledge Analysis (WEKA) for analysis. This analysis used instances totaling 1006 from PhishTank, whereby four classification algorithms were used to run several experiments. The effectiveness of the features used was measured by classification accuracy. In the experiments using decision tree algorithms, the authors noted a detection rate of 83% of the phishing sites. The authors further pointed out that when this algorithm is coupled with pre-processing, detection accuracy is significantly improved and would be used to make a very good detection model.

A Machine Learning covering algorithm, which goes by the name, *Enhanced Dynamic Rule Induction (eDRI)*, is among the first algorithms to be used as an anti-phishing solution Thabtah F., Qabajeh I., Chiclana F., (2016). To process the datasets, this Covering algorithm uses frequency and Rule strength as the two major starting points. eDRI only stores '*strong'* features of the datasets if their frequency exceeds the minimum frequency threshold after scanning all the presented datasets. The stored features are incorporated in the rule, whereas all other values are gotten rid-off in this first process. *eDRI* removes its training cases, and then strong feature occurrences are updated to signify the inexistence of the instances. This process is done when a rule has been realized. This means *eDRI* removes its instances and retains strong features. This means *eDRI* removes features by itself, providing better controllable phishing models. In order to determine *eDRI* reliability, experiments were carried out on multiple phishing websites. 11,000 websites were collected for these experiments. *eDRI* showed better results than decision trees and other covering algorithms regarding phishing detection rate. A technique called trial and error Neural Networks which uses Machine Learning, has been condemned due to its time consumption Mohammad, R. M., Thabtah, F. & McCluskey, L., (2013). For this technique to be effective, a person knowledgeable about the domains is needed during the tuning phase. The elimination of trial and error was proposed but adopted a better self-structuring classification Thabtah F., Mohammad R., McCluskey L.,(2016). The authors improved the phishing model by improving the learning rate and other parameters and later adding new neurons to the layer that is not visible. This means the features used to build the model are updated during the process of classifier model design.

According to Mohammad R., Thabtah F., and McCluskey L.,(2015), using a dynamic Neural Network model aimed to identify phishing cases from the dataset. Different dataset sizes were used to achieve this, i.e., 100, 200, 500, and 1000. These experiments showed improved predictions in comparison to Bayesian networks and decision tree techniques.

Since phishing attackers constantly update their phishing techniques, there was a need to develop a more resilient model based on the previous training results, Thabtah.F., Mohammad. R., & McCluskey L., (2014). This aimed to develop a self-learning model to counter the ever-changing techniques used by phishers. The above Neural network algorithm tracks the model's performance by using smart decisions on the results of the validation dataset. The training phase goes as follows; when the error is below the minimum, the algorithm saves up the weights and proceeds with the process. However, if the fault exceeds the lower limit, the algorithm goes further without saving any weights. Parameters can be frequently updated without waiting for the model to be completely built. This experiment revealed that the Neural network model resulted in superior prediction rates compared to traditional techniques like C4.5 and probabilistic.

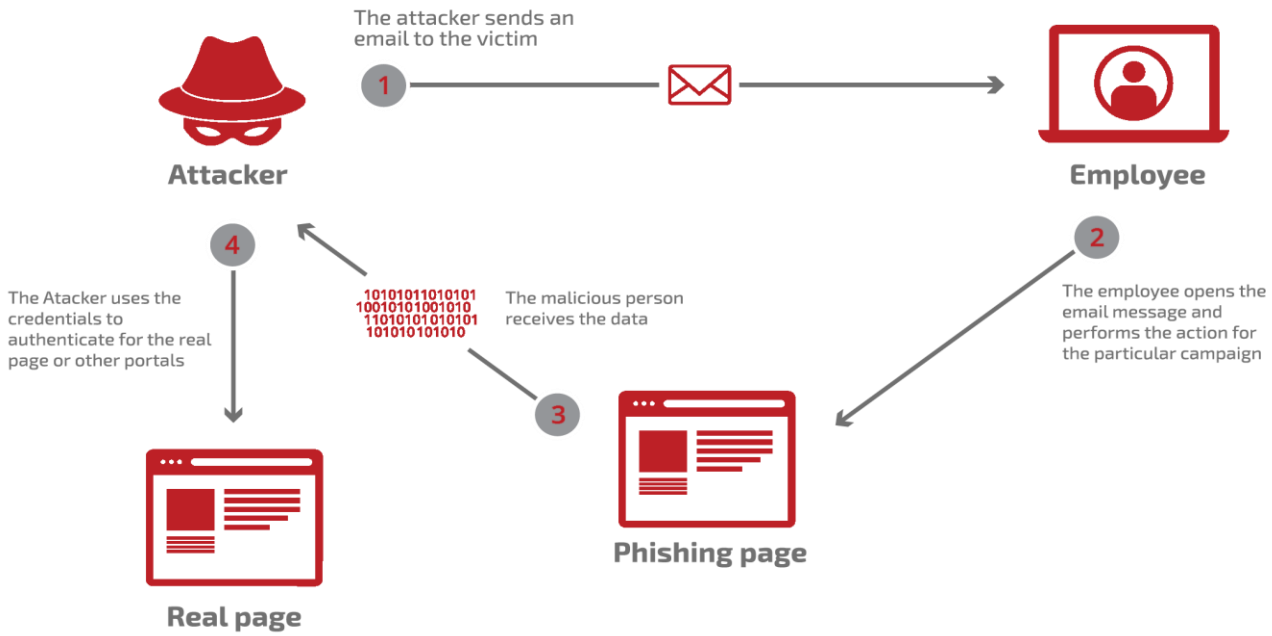


Figure 2.2 Phishing attack

2.5 Conceptual Model

The project used an ensemble machine-learning technique and fifteen features to develop a better classifier model. The features used are as described below.

2.5.1 Features used

This section describes the phishing features that our classifier used. The features were extracted and identified from the literature and formed a combination of features that effectively classify phishing and benign emails. In this project, we used 15 features identified from different literature commonly used by phishing attackers.

IP-Based URLs

Legitimate websites usually have their names on the URL. A case like http://www.mytours.com/ informs the user that someone will visit a website with the domain *mytours.com.* Attackers usually mask their identity by replacing the domain name with an IP address, e.g., http://42.56.100.21/login.asp. By doing this, malicious actors can escape detection by using IP-based URLs, which indicates a possible phishing attack. This discussed feature is identified in the literature Fette. I, Sadeh. N, and Tomasic. A, (2017).

LINK Text mismatch and "HREF" Attribute

A link to another website is usually defined by using an HTML <a> anchor tag. "href" attribute allows a user to visit another website by describing the location of the second website. The content is displayed on the browser when the user clicks the link. This link is in the form of a href="URL Address"> link text . The link text can be plain text, an image or any element. If there is a match between the link text and the pointed website, the website could be phishing. Two items are checked for mismatch, i.e., link text and href attribute for all the emails. A positive Boolean is recorded when a mismatch is found on these emails.

Link Text of Hyperlink

Phishing emails exhibit certain characteristics on the links that make the emails qualify to be phishing emails. The emails contained certain words like *click here, log in or update*.

Emails are checked for the presence of these words, and a Boolean value is recorded if these words are found or not.

Dot contained in Domain Name

According to Emigh, A.,(2016), a legitimate domain name should contain less than three dots. If the number of dots in the URL exceeds three, a binary value of 1 is noted to assist in phishing features.

HTML Email

MIME standards define every email. MIME standards define what makes up the email and its components. The components are categorized into two types, i.e., *text/plain and text/html*. These are the content-type according to Fette. I, Sadeh. N, and Tomasic. A, (2017), an email could be a phishing email if it has a "text/html" property. They argued that using HTML links is easier to achieve phishing attacks.

Use of JavaScript

JavaScript is a scripting language that is used to perform a particular action. JavaScript is either used in the body of the email using special tags denoted by <script> or can be used on a link using a tag called anchor <a>. Malicious actors make use of JavaScript language to evade detection by hiding information from users with the use of JavaScript. If an email contains a JavaScript code, it is classified as a potential phishing email, Fette I., Sadeh N. & Tomasic A.,(2017).

Links found in an Email

The sum of links in an email is registered to detect phishing emails. An email containing many links is a probable candidate for a phishing email. Phishing emails usually have links to external websites that redirect users to the attackers' websites, Yuan, Y., & Zhang, N., (2012).

Email Domain Names

The sum of unique domain characters is extracted for comparison with the referenced URLs. The incidences are recorded, and the value is used as a feature for detecting a phishing email. Each occurring unique domain name is recorded once, and any subsequent occurrence is discarded. It is therefore believed that if an email contains multiple domain names, it is a potential phishing email.

Body-From Domain Match

Domain names form a crucial part of phishing detection. This is because the domain identity of the sender and those in the body of the email should match if an email is to be classified as genuine. A match is performed on the sender's domain name and that of the extracted domain names from the email. The *"From"* field gives the sender's domain name and is compared with our test dataset for a match. If there is a disparity between the comparisons, this suggests it could be a potential phishing email, Altaher, A., Wan, T.C., ALmomani, A., (2012).

Word List

Phishing emails usually contain some occurring words which can be used as phishing detection features. These words were categorized into six categories, each of which was used as a single detection feature. This translates to having six different phishing features. Every word is counted in each category, and duplicates are discarded (normalized). These categories are;

a) Confirm; Update
b) Customer; Client; User
c) Restrict, Suspend, Hold
d) Notification, Account, Verify
e) Password, Click, Username, Login
f) Social Security; SSN

2.5.2 Email datasets/classifier/ parser/Sanitiser/Vectoriser

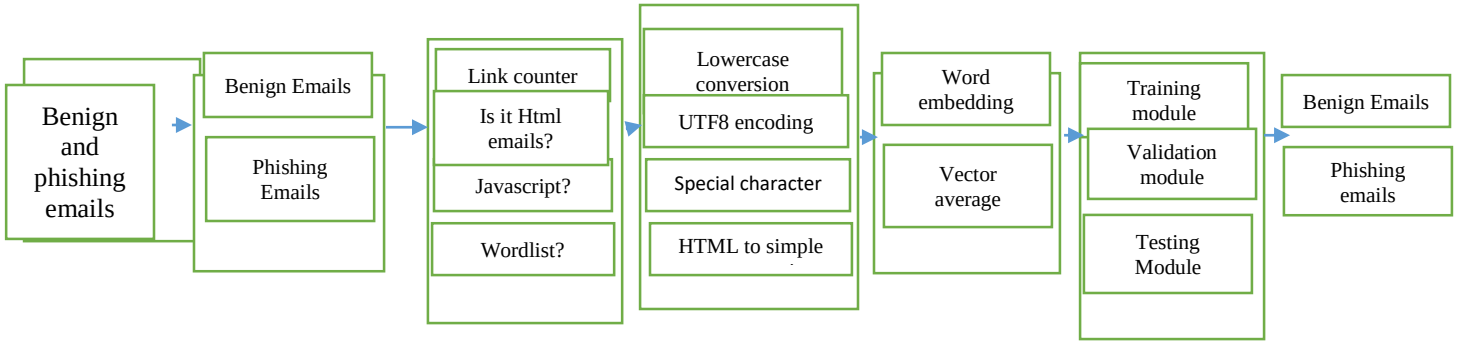


Figure 2.3 Proposed classifier model

CHAPTER THREE

METHODOLOGY

3.1 Introduction

The chapter accounts the methodology and design, testing, training and validation of the phishing classifier model.

3.2Methodology

3.2.1 Model Life cycle

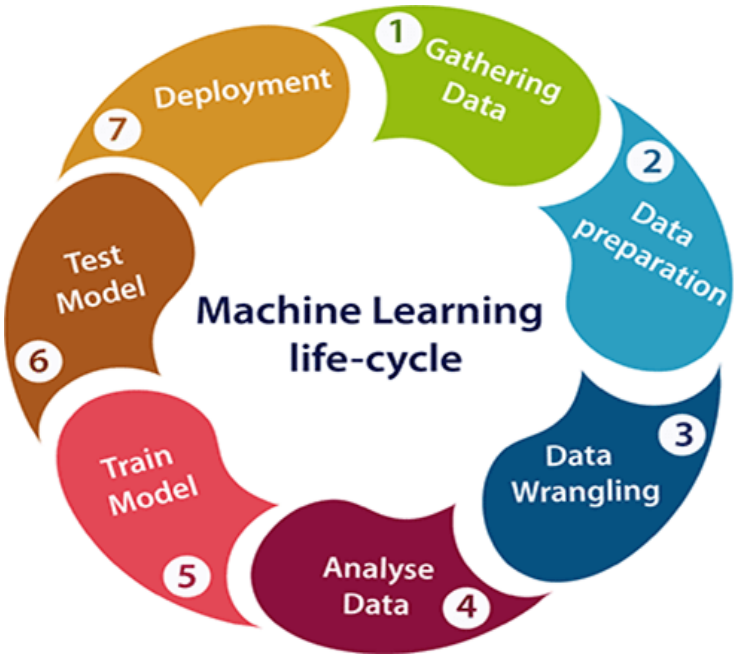


Figure: 3.1: Model development stages

Our machine-learning project used 7 stages, as shown above, to develop the phishing classifier model.

Data Gathering

This is the initial step of the machine-learning model development. Information was assembled from different hotspots for both harmless and phishing messages. The quality and amount of the information gathered decided the effectiveness of the result of the classifier model, meaning the more information utilized, the more exact the results was realized. For a model to attain the best accuracy, it needs a vast amount of data for training and testing. This data must be accurate, clean and relevant to ensure debugging the model is made easier later on.

Data Preparation

Data should be prepared for further steps after it has been collected. Here the data is put together and then randomized. This step is further divided into two categories; data exploration and data pre-processing. At the data exploration stage, we understood the nature of data, characteristics, format and quality to find correlations, general trends and outliers.

Data Wrangling

This is the process of cleansing and changing over the raw data into a more useful format by selecting the variable to use and modifying data into the best format for good analysis for the next steps. Cleaning data is important to achieve a quality model Information is cleaned to wipe out missing qualities, copy information, invalid information and commotion.

Data Analysis

The cleansed data is then passed to the next stage for analysis, which comprises of selecting analytical techniques, building models and reviewing the outcome. Here we select the AI method like arrangement, grouping, relapse, bunch examination or affiliation. The model is fabricated utilizing the all-around pre-arranged information and afterward assessed.

Train Model

This is the step where the model is developed to make better its performance for good results. We used the already acquired dataset. Training is important so the model can understand several patterns, rules and features.

Test Model

Once the model development has been done, it is then tested. Here, we check the model's accuracy by using a test dataset. Testing determined the model's accuracy per the project's objectives.

3.2.2 Research methods

Machine learning is made up of the training phase and the testing phases. We used two datasets to train our model; benign and phishing emails.

We obtained the dataset sets from Alexa for genuine (Benign) emails and PhishTank for luring (phishing) emails. We used a combination of models with an ensemble model called Random Forest (RF) to increase detection accuracy. EMBER (Open source threat data training set), IBM Watson, Existing Anti-Phishing solutions like Spam Assassin and PhishTank and Alexa for data sets are tools to assist in data modeling. We used Python libraries like *sci-kit-learn, pandas,*

numpy, matplotlib, and Java. This research used a quantitative research method to answer the question of the model's accuracy.

3.3 Training, Testing and Validation

To train and test our classier, we used a method called *10-fold cross-validation*. In this method, the training dataset was be prepared by classifying the dataset into 10 parts. Out of the 10 parts, 9 was used to train our classifier model, and the results obtained from this training was used to validate the 10^{th} group of the dataset. The process was repeated 10 times so that all ten parts was used as training and testing data. The cross-checking technique ensures that the information used for training and testing are very different. In Machine Learning projects, this method of 10-fold cross-validation has proven to produce a very good error estimate of the classifier model.

3.3.1 Training the module

Regarding the training module, three constituents were used: Input Matrix, Target Matrix and Fitness Network. These three components are used consecutively to train the classifier model better and increase the detection rate.

Input Matrix

At this stage, the model uses genuine emails from the Alexa dataset and phishing emails from PhishTank during the training stage of development. The first stage with these email datasets is to *parse* the emails by *email parser*. Then the emails are sanitized by what is known as *email sanitiz*er; lastly, the emails are vectorized by what is known as *email vectorizer*. This research had x * 5 as the logical matrix, indicating 10,000 rows, and the other part of the matrix is 5, meaning 5 columns. 10,000 means there was a total of 10000 emails dataset, with 4000 being benign and the other part of 6000 being known phishing emails.

Every email had have fifteen features with a vector size of 15.

Target Matrix

At this stage, the decisions for all benign and phishing emails are found here. The emails stored in the input matrix each produce decisions found in this matrix. In this project, we had a 10000 * 1 matrix meaning that 10000 was the total number of emails, whereas l was the vector size. The emails carry 0 or 1, where 0 denotes a benign email while 1 represents a phishing email.

Fitness Network:

This is where model formalization and testing takes place. The input and target matrix data are utilized in training, formalizing and testing. In this project, 15% was used for validation, 15% for testing and 70% for training.

3.3.2 Model Validation and Testing

The validation and testing are the last stage in the model development. At this stage, two matrixes are used: Sample and output.

Sample Matrix:

This has data from the input matrix, which is usually sample data. After the model is trained, it uses data from the sample matrix, which is used during the testing stage. In our project, this matrix is an m*5 matrix containing sample data from the input matrix.

Output Matrix:

Data from the sample matrix produces data that is found in this matrix. After training the model, it stores output values in the out matrix. This project represents this by an n * l matrix which contains output data for emails represented in the sample matrix. Using the emails in the sample matrix, the trained model predicted if an email is benign or phishing. The output matrix stored these predictions and was used to evaluate the performance of the Random

Forest algorithm. To 8achieve our objectives, we plan to use the scikit learn framework to develop, train, validate and then test our classifier model.

The scikit-learn *KFold* class automatically implemented k-fold cross-validation on the given data set. We intend to use 10-fold cross-validation.

3.3.3 Data source

The experimental data was collected from two different online sources, whereby one dataset contained benign URLs while the other contained phishing URLs. To collect data for the benign URL dataset was collected from Àlexa, which is a free, open-source data repository site that ranks URLs based on their popularity and non-malicious. The phishing email was retrieved from the PhishTank website repository. This is a free community website that enables users all over the world to submit, confirm, analyze and share phishing URL data, PhishTank (2016). The testing datasets was prepared for testing by cleansing and ensuring no duplicates. This results in clean training and testing datasets. After the dataset preparation, the training dataset comprised of 4,000 URLs, 3,000 from the benign dataset and 1,000 from the malicious set. Moreover, the testing dataset consisted of 6,000 URLs, 2,000 from the benign dataset and 4,000 from the malicious set. To realize the best results, all URLs were picked randomly, apart from any URLs that were selected in the testing dataset that don't contain the sets in the training set. The next stage was to extract various features from the URLs that have been prepared and cleaned. To realize quality among features, numeral values was normalized to be between 0 and 1. In this regard, the features are counts and binary representing values of specific entities within the URL.

CHAPTER FOUR

RESULTS

4.1 Introduction

This chapter contains detailed findings and a discussion of data analysis and the result of the study. Having developed the phishing detection model, evaluating its performance and improving it whenever necessary was important. Here, the experiment is illustrated together with outcomes, and the evaluation was set into different phases as shown below;

Phase 4.1: shows the data set, which is a gathering of 6,000 messages (genuine and phishing)

Phase 4.2: portrays the apparatuses utilized. The WEKA device tests the information utilizing the inherent AI calculation.

Phase 4.3: shows the results of the testing.

4.2 Data Set

The current data set consisted of 6000 emails, with 3000 of them being phishing emails sourced from PhishTank, and 3000 legitimate emails obtained from Alexa. This data was collected with the intent of providing a comprehensive overview of the current phishing landscape and offer a basis for further data mining. The Spam Assassin has two different email types: those easily identified as legitimate and those that are hard to differentiate from spam. The hard-to-tell emails, while still legitimate, need a lot extra checking to ensure they are not actually spam.

4.2.1 Describe the Dataset

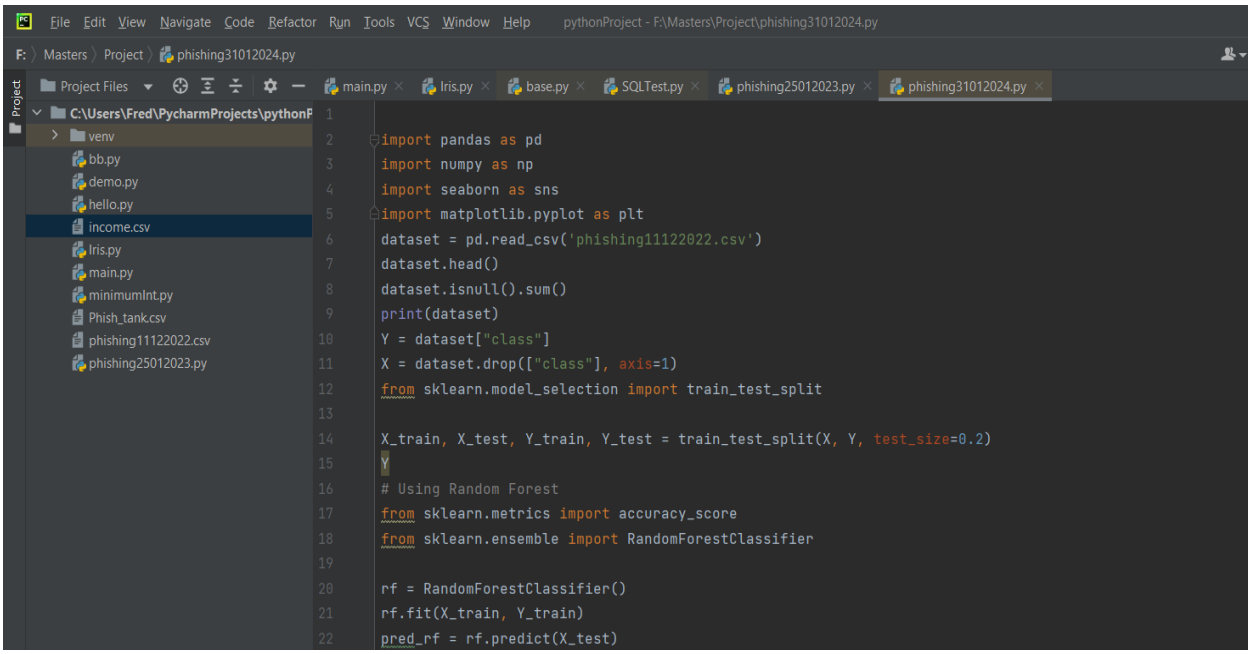


Figure 4.1: Coding of the phishing classifier

Figure 4.1 illustrates the information contained within the Phishing Website dataset. Initially, the coding demonstrates the set of data that is being utilized, the Phishing Website dataset. The information is mostly contained within a CSV file, making it effortless for the Kaggle platform to interpret and execute the information set. In this instance, the dataset contains 15 columns and 11055 rows, with three values in the dataset: -1 representing phishing, 0 representing suspicious, and 1 representing legitimate.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Index	UsingIP	LongURL	ShortURL	Symbol@	Redirectin	PrefixSuffi	SubDomai	HTTPS	DomainRe	Favicon	NonStdPo	HTTPSDon	RequestUF	AnchorUR	LinksInScri	ServerForr
2	0	1	1	1	1	1	-1	0	1	-1	1	1	-1	1	0	-1	-1
3	1	1	0	1	1	1	-1	-1	-1	-1	1	1	-1	1	0	-1	-1
4	2	1	0	1	1	1	-1	-1	-1	1	1	1	-1	-1	0	0	-1
5	3	1	0	-1	1	1	-1	1	1	-1	1	1	1	1	0	0	-1
6	4	-1	0	-1	1	-1	-1	1	1	-1	1	1	-1	1	0	0	-1
7	5	1	0	-1	1	1	-1	-1	-1	1	1	1	1	-1	-1	0	-1
8	6	1	0	1	1	1	-1	-1	-1	1	1	1	-1	-1	0	-1	-1
9	7	1	0	-1	1	1	-1	1	1	-1	1	1	-1	1	0	1	-1
10	8	1	1	-1	1	1	-1	-1	1	-1	1	1	1	1	0	1	-1
11	9	1	1	1	1	1	-1	0	1	1	1	1	1	-1	0	0	-1
12	10	1	1	-1	1	1	-1	1	-1	-1	1	1	1	1	-1	-1	-1
13	11	-1	1	-1	1	-1	-1	0	0	1	1	1	-1	-1	-1	1	-1
14	12	1	1	-1	1	1	-1	0	-1	1	1	1	1	-1	-1	-1	-1
15	13	1	1	-1	1	1	1	-1	1	-1	1	1	-1	1	0	1	1
16	14	1	-1	-1	-1	1	-1	0	0	1	1	1	1	-1	-1	0	-1
17	15	1	-1	-1	1	1	-1	1	1	-1	1	1	-1	1	0	-1	-1
18	16	1	-1	1	1	1	-1	-1	0	1	1	-1	1	1	0	-1	-1
19	17	1	1	1	1	1	-1	-1	1	1	1	1	-1	-1	0	-1	-1
20	18	1	1	1	1	1	-1	-1	1	-1	1	1	1	1	0	0	-1
21	19	1	0	-1	1	1	-1	0	1	-1	1	1	1	1	0	0	-1
22	20	1	0	1	1	1	-1	0	1	1	1	1	-1	-1	0	-1	-1
23	21	1	1	1	1	1	-1	-1	-1	-1	1	1	-1	1	0	0	-1
24	22	1	1	1	1	1	-1	1	0	-1	1	1	1	1	0	0	-1
25	23	1	-1	-1	-1	1	-1	1	1	-1	1	1	-1	-1	0	0	-1
26	24	1	-1	1	1	1	-1	0	1	-1	1	1	1	1	1	0	-1
27	25	1	-1	1	1	1	-1	0	-1	1	1	1	-1	-1	-1	-1	-1
28	26	1	-1	-1	1	1	1	-1	1	1	1	1	1	-1	1	0	-1
29	27	1	-1	-1	1	-1	1	-1	1	-1	1	1	1	1	1	0	-1

Figure 4.2 Sample of the Dataset 15 Feature

4.2.2 Dataset Split for Training and Testing

```
dataset.head()
dataset.isnull().sum()
print(dataset)
Y = dataset["class"]
X = dataset.drop(["class"], axis=1)
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)
Y
```

Figure 4.3: Split for Training and Testing

Figure 4.3 influences 70% of the phishing site dataset to prepare and fit the boundaries, while the excess 30% is utilized to assess the model's adequacy. The training dataset is borrowed to

hone the parameters and the testing dataset is utilized to measure the performance of the model.

4.2.3 Add the Algorithms

```
Y = dataset["class"]
X = dataset.drop(["class"], axis=1)
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)
Y
# Using Random Forest
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier()
rf.fit(X_train, Y_train)
pred_rf = rf.predict(X_test)
print(accuracy_score(pred_rf, Y_test))
from sklearn.metrics import classification_report, confusion_matrix

classification_repo = classification_report(pred_rf, Y_test)
confusion_mat = confusion_matrix(pred_rf, Y_test)
print(confusion_mat, "\n", classification_repo)
print('Random Forest Accuracy:',accuracy_score(pred_rf, Y_test))

# Using Decision Tree
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
dt.fit(X_train, Y_train)
pred_dt = dt.predict(X_test)
print(accuracy_score(pred_dt, Y_test))
from sklearn.metrics import classification_report, confusion_matrix
```

Figure 4.4: The algorithms

The evaluation of the performance of the classifying the phishing websites is demonstrated in Figure 4.4 through the use of three distinct machine learning algorithms: Random Forest,

Decision Tree classifier , Adaboost and Naïve Bayes. This provides a comprehensive analysis of the Classification Accuracy (CA) of each model in terms of effectiveness and efficiency.

4.3 Tools

To evaluate the compatibility of the file, it was converted to a CSV format and tested using the five algorithms selected by the WEKA tool. The results of this experiment determined whether the file can be used with the WEKA tool or not. Weka is a powerful set of machine learning algorithms designed to tackle a variety of data mining tasks. It provides a range of tools to pre-process, classify, regress, visualize, and cluster data. These algorithms can be used directly on the dataset or called from Java code, making it ideal for developing new machine learning approaches. With its perplexing capabilities and high bustiness. Weka is a powerful tool for data mining, offering a broad variety of algorithms to help with any data mining task. This software provides users with the ability to analyze and uncover hidden patterns in large datasets. With Weka, users can quickly and easily explore, visualize, and manipulate data, and ultimately make more informed decisions, Weka, (2016).

4.4 Experimental results

This section conducts various experiments in different scenarios, evaluate experiments and results using various measures, compare the performance of several experiments, and highlight the results.

The phishing classification model was implemented through generation of the Random Forest Classifier, Decision Tree Classifier and the AdaBoost algorithm. This project aims to evaluate the use of ensemble methods against other algorithms and determine which method is better for

ranking phishing and non-phishing websites. These algorithms are also generated using Python.

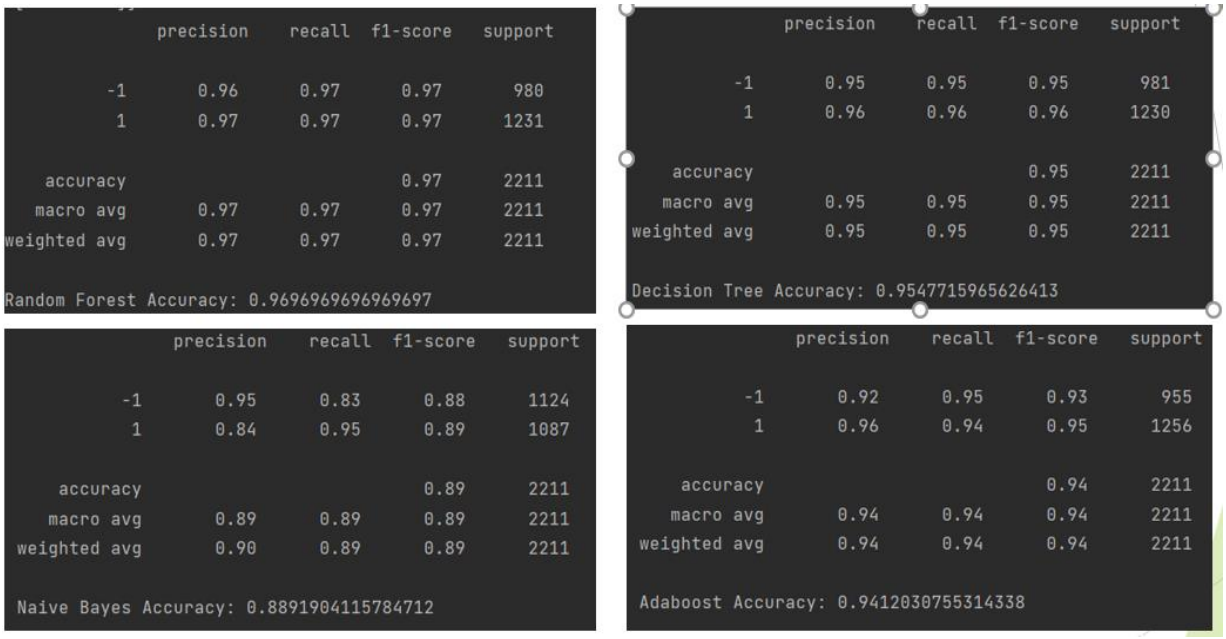


Figure 4.5: Comparison of Accuracy of four Algorithms

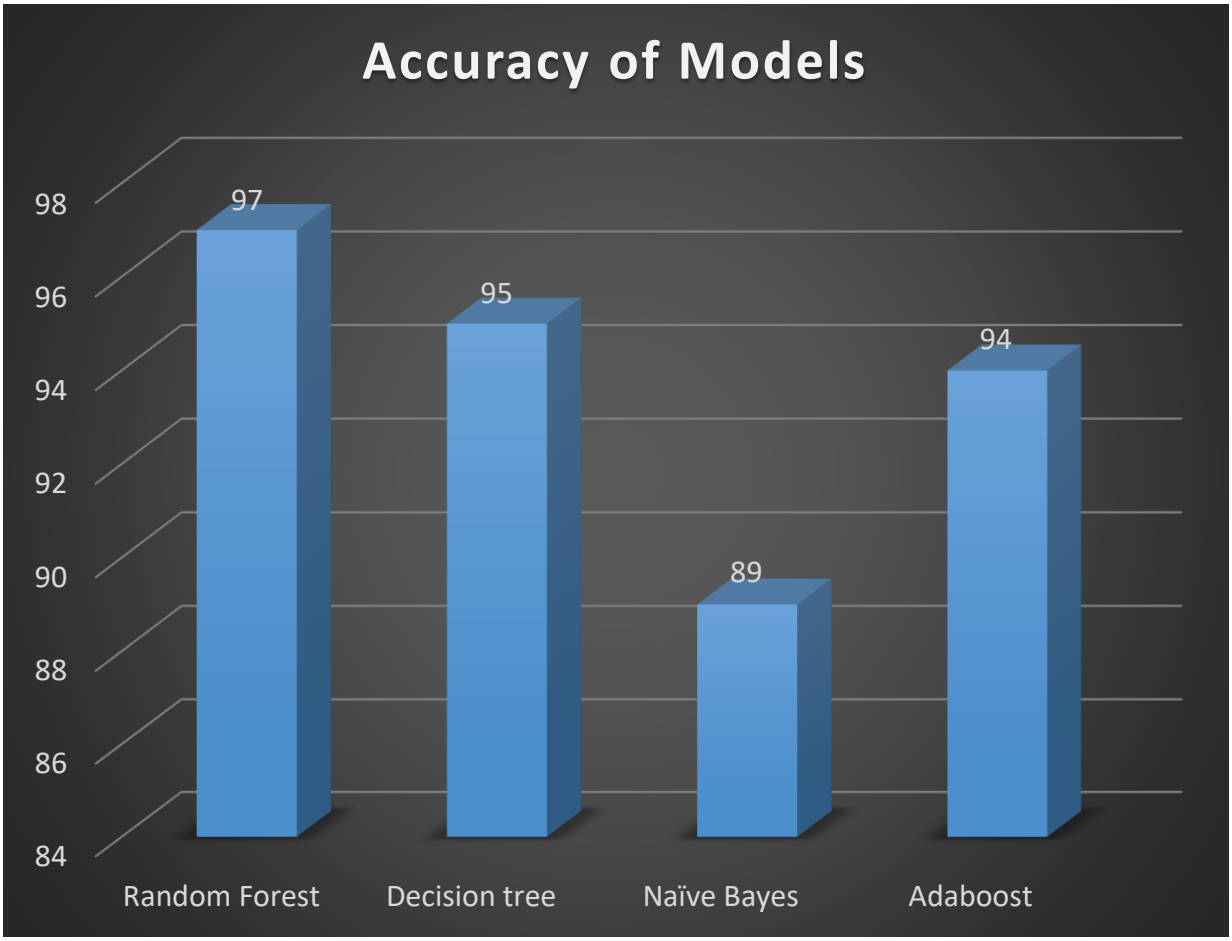


Figure 4.6: Bar plot Accuracy of four Algorithms

Table 4.1: Accuracy of four Algorithms

Model	Accuracy	Precision	Recall	F1 score
Random Forest	**97%**	**97%**	**97%**	**97%**
Decision tree	**95%**	**95%**	**95%**	**95%**
Naïve Bayes	**89%**	**90%**	**89%**	**89%**
Adaboost	**94%**	**94%**	**94%**	**94%**

4.5 Summary

The machine-learning algorithm has complete functions, the phishing email classification accuracy is higher, and the random forest algorithm has higher accuracy. This shows that the algorithm is well suited for data classification, especially in the phishing email dataset. This shows that the ensemble algorithm performs better in phishing classification because it uses multiple decision models.

CHAPTER FIVE

DISCUSSION

5.1 Introduction

This chapter deals with discussion, limitations, suggestions and improvements. The results of this project were based on expected results and were instrument tested. Figure 4.6 and Table 1 illustrates a difference in algorithms and accuracy. Indicating that the Random Forest classifier gives the biggest accuracy, which is 97%, then the Decision Tree 95%, AdaBoost 94% and Naïve Bayes 89%. This output demonstrates the accuracy percentage, whereas the training and testing sets are identified with different parameters in the dataset. From the results, it shows ensemble model has higher classification accuracy. Therefore, ensemble performed better in classification of genuine and phishing emails.

Considering other metrics like precision, recall and f1 score, Random Forest still performs better than other models.

Contributions are collected through system goals. The challenges are evaluated by studying the completeness of the model or the challenges and difficulties encountered during the development process of the classifier.

Phishing emails are now a norm in recent years. Phishing is when the victim sends an email requesting key information from the user, which is sent directly to the phisher. Therefore, tracking these emails is necessary. There are numerous innovations to distinguish phishing messages. In any case, they all have restrictions, for example, low exactness, the substance might be like authentic messages and in this manner can't be recognized, and the identification rate should be higher; thus, they have high bogus positives and high misleading negatives. This

study evaluated the accuracy of phishing email detection through the use of manual selection feature and also the use of automatic feature selection of three classification algorithms that have high detection rates.

At the end of the process, the two scenarios are compared to determine the method that yields better results in terms of detection rates.

For manual attribute selection, 15 email attributes were chosen and divided into four categories based on email structure (body attributes, header attributes, URL attributes and Java script attributes with external attributes). The results indicate that the body group has the highest accuracy rate in detecting phishing emails, reaching 91.16%.

On the other hand, all but one of the four groups were tested together for accuracy each time. The results indicate that the highest accuracy rate, 97, is achieved if the URL attribute group is removed from all the attributes.

Using auto-selection of the project testing, the accuracy was tested on three sets of auto-selected features, which are generated by the system. The results showed a deviation in accuracy between the three categories, with the highest group being the third one achieving 98% precision.

5.2 Conclusions

After completion of the research, we can draw these conclusions;

There are various methods attackers use to lure users on the interment to click malicious links so that that accounts can be compromised. These methods are creating websites that look like genuine ones, creating emails that mimic an ally of the recipient or downloading an email attachment. The use of ensemble model and more email features gives higher accuracy in

phishing email detection as opposed to the traditional single model and single email feature like domain url.The developed classifier model can detect a phishing email with accuracy of 97%

5.3 Recommendation

Based on the findings of the research, phishing is a real threat as a result of a user clicking a link and inputting their personal details on malicious websites. There is need to ensure the page requesting for personal data is secure and trusted before supplying it with the data. Never input personal data on an unverified page to avoid exposing one's personal data. More so, users need to use anti-phishing tools on the gadgets they use to visit sensitive pages, which require personal data to avoid falling into prey of the hackers. It is imperative to use phishing classifier tools that have high detection accuracy of more than 99.9%. The best tools use ensemble AI model and use many features of the email body to develop a classifier model.

5.4 Future Research

More work is needed for future feature selection techniques since selection techniques still need to be refined to cope with new techniques that anglers develop over time. Thusly, we propose to obtain another mechanized device to separate new elements from new crude messages to improve phishing email location precision and adapt to the extension of phishing methods.

REFERENCES

Abdelhamid, N., Thabtah, F. (2014). Associative Classification Approaches: Review and Comparison. Journal of Information and Knowledge Management (JIKM). Vol. 13, No. 3 (2014).

Aburrous, M., Hossain, M., Dahal, KP & Thabtah, F. (2010). Experimental Case Studies for Investigating E- Banking Phishing Techniques and Attack Strategies. Journal of Cognitive Computation, Springer Verlag, 2 (3): 242-253.

Afroz, & Greenstadt, R. (2011). PhishZoo: Detecting Phishing Websites by Looking at Them. In Fifth International Conference on Semantic Computing (September 18- September 21). Palo Alto, California USA, 2011. IEEE.

Akinyelu, A. A. & Adewumi, A. O. (2014). Classification of phishing emails using random forest machine learning technique. Journal of Applied Mathematics, vol. 2014, Article ID 425731, 6 pages, 2014.

Altaher, A., Wan, T.C., ALmomani, A.,(2012). "Evolving fuzzy neural network for phishing emails detection," Journal of Computer Science, vol. 8, no. 7.

APWG Phishing Attack Trends Reports https://www.antiphishing.org/resources/apwg-reports/ , 2018

Basnet, R., Mukkamala, S., Sung, A.H. (2008). Detection of phishing attacks: A machine learning approach (2008) Soft Computing Applications Industry, pp. 373-383.

Bayesian network classifiers in Weka. (Working paper series. University of Waikato, Department of Computer Science. No. 14/2004). Hamilton, New Zealand: University of Waikato.

Behdad, M., French, T., Bennamoun, T. and Barone, L. (2012) "Nature-inspired techniques in the context of fraud detection," IEEE Transactions on Systems, Man, and Cybernetics C

Bouckaert, R., (2004). Bayesian network classifiers in Weka. (Working paper series. University of Waikato, Department of Computer Science. No. 14/2004). Hamilton, New Zealand: University

Brown, S., Ofoghi, B., Ma, L. & Watters, P., (2017)."Detecting phishing emails using hybrid features," Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing (UIC-ATC '17), IEEE, Australia.

Cranor, L.F, J. I. Hong and Y. Zhang, (2016) "Cantina: a content-based approach to detecting phishing websites," 16th International World Wide Web Conference (WWW '07), Canada.

Cutler, A. and Breiman, L., (2007) “Random forests-classification description,” Department of Statistics Homepage

Emigh, A., (2016). “Phishing attacks: information flow and chokepoints,” in Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft, USA.

Fette I., Sadeh N. & Tomasic A. (2017). Learning to detect phishing emails. Proceedings of the 16th international conference on the World Wide Web. 649-656.

Freund Y. & Schapire R. E. (1997). A decision-theoretic generalization of online learning and an application to boosting. Journal of Computer and System Sciences, 55(1):119-139, 1997.

Gaines, B.R. & Compton, J.P. (1995). Induction of Ripple-Down Rules Applied to Modeling Large Databases, Intell. Inf. Syst. 5(3):211-228

Gupta, M., Prakash, P., Kompella, R.R. and Kumar, M. (2015) "PhishNet: predictive blacklisting to detect phishing attacks," IEEE Conference on Computer Communications.

Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I. (2009) The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

Han, W., Cao, Y. & Le, Y. (2015). “Anti-phishing based on automated individual white-list,” 4th ACM Workshop on Digital Identity Management (DIM), pp. 51–59, ACM USA

Holbrook, M., Kumaraguru, P., Downs, J., Cranor, L.F. and Sheng, S. (2010, April) "Who falls for phish?

Holte, R.C. (1993). Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. Machine Learning, 11, pp 63-90.

Huber M, Mulazzani M, Leithner M, Schrittwieser S, Wondracek G, Weippl, E. (2011) Computer Security Applications, 27th Annual Computer Security Applications Conference

Khonji, M, Jones, A and Iraqi, Y. (2013) "Phishing detection: a literature survey," IEEE Communications & Surveys Tutorials.

Ledesma, R., Chou, N., Mitchell, J.C. & Teraguchi, Y. (2014). “Client-side defense against web-based identity theft,” 11th Annual Network & Distributed System Security Symposium, USA.

Mitchell, T.M. (1997). Machine Learning, McGraw-Hill, New York, NY, USA

Mohammad R., Thabtah F., McCluskey L. (2015B) Phishing websites dataset. Available: https://archive.ics.uci.edu/ml/datasets/Phishing+Websites Accessed January 2016.

Mohammad R., Thabtah F., McCluskey L., (2014A) Predicting Phishing Websites based on Self-Structuring Neural Network. Journal of Neural Computing and Applications, 25 (2). pp. 443-458. ISSN 0941-0643. Springer.

Mohammad, R. M., Thabtah, F. & McCluskey, L. (2013). Predicting Phishing Websites using Neural Network trained with Back-Propagation. Las Vegas, World Congress in Computer Science, Computer Engineering, and Applied Computing, pp. 682-686.

Nargundkar, S., Tiruthani, N. & Yu,W.D. (2017). “PhishCatch—a phishing detection tool,” 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC '17), USA

Nazif, M., Ryner, B. and Whittaker, C (2010)"Large-scale automatic classification of phishing pages," 17th Annual Network & Distributed System Security Symposium (NDSS '10), The Internet Society, USA.

Platt J. (1998). Fast training of SVM using sequential optimization,(Advances in kernel methods support vector learning, MIT Press, Cambridge, 1998, pp. 185-208

Qabajeh I., Thabtah, F., Chiclana, F. (2015) Dynamic Classification Rules Data Mining Method. Journal of Management Analytics. Volume 2, Issue 3, pp. pages 233-253. Wiley.

Quinlan, J. (1993). C4.5: Programs for machine learning. San Mateo, CA: Morgan Kaufmann.

Sadeh, N., Fette, I & Tomasic, A. (2017). “Learning to detect phishing emails,” 16th International World Wide Web Conference (WWW '17), Canada.

Smadi, S., Aslam, N., Zhang, L., Alasem, R. & Hossain, M.A., (2015). Detection of phishing emails using data mining algorithms.

Strobel,S., Glahn,S., Moens,M.F., & Bergholz, A.(2010). “New filtering approaches for phishing email,” Journal of Computer Security, vol. 18, no. 1, pp. 7–35,

Sung, A.H., Basnet, R. and Mukkamala, S. (2008). “Detection of phishing attacks: a machine learning approach,” in Soft Computing Applications in Industry, Germany

Tan C.L., Chiew K.L., Sze S.N. (2017) Phishing Webpage Detection Using Weighted URL Tokens for Identity Keywords Retrieval. In: Ibrahim H., Iqbal S., Teoh S., Mustaffa M. (eds) 9th International Conference on Robotic, Vision, Signal Processing and Power Applications. Lecture Notes in Electrical Engineering, vol 398.Springer, Singapore

Thabtah F., Mohammad R., McCluskey L. (2016B) A Dynamic Self-Structuring Neural Network Model to Combat Phishing. In the Proceedings of the 2016 IEEE World Congress on Computational Intelligence. Vancouver, Canada.

Thabtah F., Qabajeh I.., Chiclana F. (2016A) Constrained dynamic rule induction learning. Expert Systems with Applications 63, 74-85.

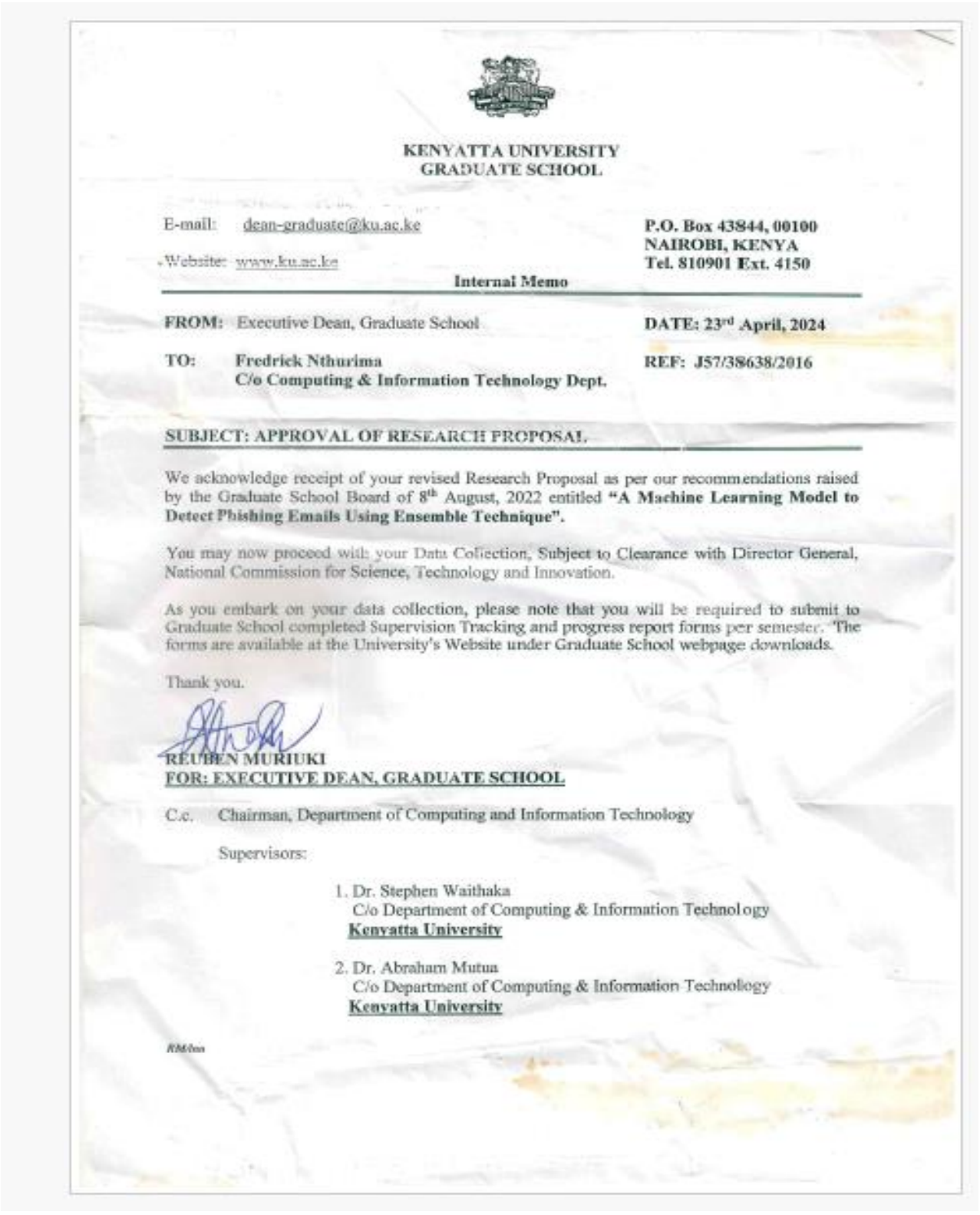
Wattenhofer, R., Burri, N. and Albrecht, K. (2015). “Spamato-an extendable spam filter system,” in Proceedings of the 2nd Conference on Email and Anti-Spam (CEAS '15), USA

Witten I. H. and Frank E. (2005). Data Mining: Practical Machine Learning Tools and Techniques.

Yuan, Y. & Zhang, N. (2012). "Phishing detection using neural network," http://cs229.stanford.edu/proj2012/ZhangYuan-PhishingDetectionUsingNeuralNetwork.pdf

Zhang,Y., Cranor,L.F., Hong,J.I, & Egelman,S. (2016). "Finding phish: an evaluation of anti-phishing toolbars," 14th Annual Network & Distributed System Security Symposium, USA.

APPENDICESAPPENDIX I: Research Approval Letter



KENYATTA UNIVERSITY
GRADUATE SCHOOL

E-mail: dean-graduate@ku.ac.ke
Website: www.ku.ac.ke

P.O. Box 43844, 00100
NAIROBI, KENYA
Tel. 810901 Ext. 4150

Internal Memo

FROM: Executive Dean, Graduate School

DATE: 23rd April, 2024

TO: **Fredrick Nthurima**
C/o Computing & Information Technology Dept.

REF: J57/38638/2016

SUBJECT: APPROVAL OF RESEARCH PROPOSAL

We acknowledge receipt of your revised Research Proposal as per our recommendations raised by the Graduate School Board of 8th August, 2022 entitled **"A Machine Learning Model to Detect Phishing Emails Using Ensemble Technique".**

You may now proceed with your Data Collection, Subject to Clearance with Director General, National Commission for Science, Technology and Innovation.

As you embark on your data collection, please note that you will be required to submit to Graduate School completed Supervision Tracking and progress report forms per semester. The forms are available at the University's Website under Graduate School webpage downloads.

Thank you.

REUBEN MURIUKI
FOR: EXECUTIVE DEAN, GRADUATE SCHOOL

C.c. Chairman, Department of Computing and Information Technology

Supervisors:

1. Dr. Stephen Waithaka
   C/o Department of Computing & Information Technology
   Kenyatta University

2. Dr. Abraham Mutua
   C/o Department of Computing & Information Technology
   Kenyatta University

RM/ms

APPENDIX II: Research Permit (NACOSTI)

REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: 773838

Date of Issue: 23/May/2024

RESEARCH LICENSE

This is to Certify that Mr.. Fredrick Nthurima Murangiri of Kenyatta University, has been licensed to conduct research as per the provision of the Science, Technology and Innovation Act, 2013 (Rev.2014) in Nairobi on the topic: A MACHINE LEARNING MODEL TO DETECT PHISHING EMAILS USING ENSEMBLE TECHNIQUE. for the period ending : 23/May/2025.

License No: NACOSTI/P/24/35747

773838

Applicant Identification Number

Director General
NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Verification QR Code

NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.

See overleaf for conditions