Email Classification Tool to Detect Phishing Using Hybrid Features

H.V. Mahesh

169321L

Faculty of Information Technology

University of Moratuwa

March 2019

Email Classification Tool to Detect Phishing Using Hybrid Features

H.V. Mahesh
169321L

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the partial fulfillment of Degree of Master of Science in Information Technology.

February 2019

Declaration

We declare that this thesis is our own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student (s)	Signature of Student (s)
H. V. Mahesh	……..
	Date ……..

Supervised by

Name of Supervisor	Signature of Supervisor
S.C. Premarathne	…….
	Date: ……...

Acknowledgements

First and foremost, I would like to express my sincere gratitude towards my supervisor, Mr. Saminda Premarathne, Senior Lecturer, Faculty of Information Technology, University of Moratuwa for his guidance, supervision, advices and sparing valuable time thorough the research project.

A special thanks goes to Mr. Indika Gunawardhana, CIO, LAUGFS Holdings Ltd for the valuable support and guidance given for the research. I would also like to thank all batch mates of my MSc IT program who gave their valuable feedbacks to improve the results of the research. Further I express my warm thanks to my family & wife for the support and encouragement they have given during this time.

Abstract

Phishing is a fraudulent attempt of trying to gather personal sensitive information such as user ID and passwords, credit card and bank account details through network. Social messaging and websites are used as medium to trigger attacks in addition to the use of emails, which is the most common and leading method currently used to perform phishing attacks. In an attack, the attacker is sending an email with a URL of the phishing website camouflaged as a legitimate source.

Nowadays phishing has become more complicated and critical problem to many organizations. The phishers can bypass the filters and rules set by anti-phishing procedures and techniques. This research build a web based phishing email detection tool using data mining classification model.

To build an efficient classification model, varieties of extracted email features have been used. These selected features can be categorized according to email header, email body, URL and Web Page Content of URL. In this model, classification accuracy will be enhanced by using these hybrid features.

This model will be used to implement the web-based tool to detect phishing emails with more accuracy even without opening the emails. This can be used as preventive and proactive technique for phishing detection.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Prolegomena

Phishing is a fraudulent attempt of trying to gather personal sensitive information such as user ID and passwords, credit card and bank account details through network. The attacker is sending an email with URL of phishing websites. Nowadays, phishing has become more complicated and critical problem to many organizations. Social messaging, web sites and emails are used by the attackers to make attacks. Among them, using emails with phishing links is the most common and leading method which is used by the phishers to mislead users. Tricky part of a phishing attack is the attackers are acting like legitimate party such as bank, social media, popular organization or web site. They influence user to provide sensitive information in to a fake web site which is similar to legitimate one. With technology improvements, phishing attacks with advanced phishing emails have been developed as well as anti-phishing techniques.

To avoid this, organizations and individuals are using various anti-phishing security methods. Although most companies and users rely on standard phishing filtering software to avoid these attacks. Most of these filters will not do the job of preventing the breach because a well-crafted advanced phishing email will not be detected by the standard filters.

1.2 Background and Motivation

Today, employees are frequently exposed to advanced phishing and ransomware attacks. According to the recent report published by the FBI [1], from October 2013 to February 2016, phishing scam affected at least $2.3 billion in damages, involving 17,642 businesses in more than 79 countries. However, phishing has become more critical and unavoidable problem of information security, so that phishers used to design rich content emails and advance features to bypass the filters which set by current anti-phishing techniques and mislead the Customers and organizations.

Most automatic phishing email detection approaches depend on email content data or URL associated with email messages but not on the URLs contained data of the message with

those. There are few phishing detection approaches which evaluate content of URL web page with some limitation. Some approaches are only focused on search engine results, Alexa ranking or blacklist data sources without focusing on heuristic method.

In this work, I will be focusing on the email header, email body, URLs and content of the URL pages. Further, Alexa ranking and blacklist data results will also be considered. There is no research found specifically focused on all the features considered in the same tool to detect phishing email. This would be the primary reason for the higher accuracy of this tool's result. To demonstrate the effectiveness of this approach, a large hybrid feature set of Phishing emails will be evaluated using several classification techniques.

1.3 Problem Statement

Phishers fool users by designing advanced email with rich html content, capable of bypassing the filters set by current anti-phishing methods. Current anti phishing tools and techniques are mainly implemented base on the features of email header, email content, URL , URL web page content or URL external details by considering individually or combine 2-3 types of features in maximum. But there is no tool found that is focused on all above feature types for phishing detection. If we can implement a tool considering all feature types, it will be more useful and more accurate. To filling the gap, this research has been focused on developing a new Phishing Classification Tool considering all listed feature types. Algorithm used in this tool will be developed using Data mining Classification Technique.

1.4 Aims and Objectives

1.4.1 Aim

To develop a new email classification tool with higher accuracy to detect phishing emails by using hybrid features.

1.4.2 Objectives

- Develop a features extraction tool.
- Build an email classification model using classification techniques.
- Evaluate the classification model.
- Develop a web-based email classification tool by applying the evaluated model.

1.5 Proposed Solution

We propose web-based system that classify emails as legitimate or phishing based on hybrid feature classification model. System facilitates the user to enter their email credential and check whether they have received phishing emails. The Tool has been built using a classification technique.

1.6 Structure of the Thesis

The overall thesis is formatted as follows:
First chapter gives an introduction of project with the background, problem, aim and objectives and solution and second chapter critically review the literature in the data mining technology in phishing with a special reference to classification technique. Third chapter is about details of data mining technology by showing it's relevance to phishing features and forth chapter present system approach with users, inputs, outputs, process and features. Fifth chapter is the design of the classification tool, while sixth chapter implementation of the tool. Seventh chapter reports on the evaluation of the solution. Finally, chapter eight concludes the solution with a note on further work.

Chapter 2

Literature Review

2.1 Introduction

This chapter critically reviews the use of data mining technique for phishing detection studies. Based on previous works this chapter focuses on feature selection and identified unsolved issues and concerns about using data mining techniques in phishing detection. Finally, we define our research problem to be addressed in the thesis.

2.2 Related work of Phishing Classification

Phishing is one of the most common types of social engineering attacks used today. According to APWG [2], the total number of unique phishing e-mails reported in 2017 1H was 499,678 and also the number of unique phishing email reports were largely consistent from month to month, except for a 21 percent spike in March 2017.

According to Jason Hongs **[3]** Phishing attacker used to three major phases for successful phishing attack. The first one is potential victims receiving a phishing email. The second step is the victim taking the recommended action in the email message, which is directed to a fake web site. The final step is the attacker use stolen information for fraud activity.

As mentioned on "A Novel Algorithm to Detect Phishing URLs study" [4], URL can be identified whether it is from a phishing website or not, using the algorithm introduced. This algorithm performs on number of URL based features, Google's updated blacklist check, Alexa Ranking and utilize google search engine results to detect phishing URLs. During the detection process, alert message will appear when the URL has been found as a possible phishing attempt. Otherwise it shows a safe to use message. Although this algorithm can be used for both known/old and unknown/new URLs, The solution is more reliable and effective only for HTTP URLs. Therefore, the researcher has noted the importance of adding other features to improve the tool by the time.

In the research "A PageRank Based Detection Technique for Phishing Web Sites" [5], a new technique has been designed and implemented based on the Google Page Rank value which is helped to detect whether the web pages are legitimate or phished sites. In this research, researcher has paid his attention on some features such as age of the domain, suspicious URL, domain contains IP address or not, number of dots and taking user personal information as input or not. And this technique has implemented only for web sites and no extension for detecting email phishing which is commonly happen in present.

In this study [6], Researcher has developed a sender- centric approach by focusing only on the sender information of the message to detect banking phishing mails. He did not focus on the content or the structure of the message. As the researcher observed, it is much difficult to conceal sender information than manipulating both content and structure of a phishing mail. There are two steps in this approach which has introduced to detect phishing banking mails. As the first step, separate the banking messages from non-banking messages and then use the algorithm to recognize the phishing messages from legitimate banking messages. So this research has not focused on all emails which receive to mail box whether they are phishing or not. It only looked at the banking mails. Also the researcher did not consider the other factors but only on sender details.

As described on "Client-Side Counter Phishing Application using Adaptive Neuro-Fuzzy Inference System" [7], intelligent techniques have been used to develop an application for detecting phishing emails in a user's mailbox. For this application, researcher has used adaptive Neuro-Fuzzy inference system (ANFIS) and intelligent hybrid technique to implement. Emails received to a user mailbox have been retrieved and checked to get the number of occurrence of each phishing indicator defined. This value has been used as the input to ANFIS. This system gives a value for each email as an output. According to this output value, emails can be identified and categorized as a type of legitimate, suspicious or phishing. The researcher has tested this application on phishing mails and proved the effectiveness and accuracy of the same. But further he described the importance of a modified structure which is suitable for number of inputs and rules proposed to get more accurate outcome.

"Detecting Phishing Emails the Natural Language Way" [8], has been presented a phishing detection scheme called PhishNet-NLP which has been implemented by utilizing available natural language based techniques and context information. An email has been checked for phishing by using the available information such as names, header, links and text of the emails. PhishNet-NLP operates between mail transfer agent and mail user agent. It checks and avoid the user from opening phishing links on the email when arriving an email. This scheme detects phishing at the level of email, rather than detecting on connected websites.

"Detecting Phishing Emails Using Hybrid Features" [9], in this paper an approach has been implemented to detect phishing emails considering hybrid features such as keyword, form, script and features of link. According to researcher's identification, classification based on email content is not enough to detect phishing, as phishing attacks have become more complicated and developed. Researcher shows the importance of considering orthographic features which reflects the sender's styles and habits. With reference to these points, researcher has extracted hybrid features from different sources to detect the phishing.

"According to Detecting phishing e-mails using Text and Data mining" [10], researcher has developed techniques based on text and data mining to detect phishing emails by focusing on the email content mainly. In here consider 2500 phishing and non-phishing emails to extract keywords from email body using text mining. Among the 23 extracted keywords, 12 keywords have been selected by using t-statistics to implement the phishing detecting tool. The researcher has tried to obtain more prediction accuracy by using less number of features using only email content keywords.

In the "Detection of Phishing Emails using Feed Forward Neural Network" [11] research, phishing detection model has been proposed based on the extracted email features appeared in the header and HTML body of an email to classify the tested emails in to phishing or not. This approach introduced to detect phishing email as quick as possible using feed forward neural network. In this research, researcher extracted 18 features from tested phishing emails to implement the approach. Then multilayer feed forward neural network has been used to classify the emails.

According to the research of "Identification and Detection of Phishing Emails Using Natural Language Processing Techniques" [12], Researcher has focused on detecting phishing mails which have no links to phishing sites. In this mail, phishing attacker set the user to reply with sensitive information by misleading him using features such as non-mentioning of the victim's name in the email, a mention of monetary incentive and a sentence inducing the recipient to reply. The researcher has identified these common features on such emails, based on the textual analysis and combined with header analysis further. Final combined evaluation has been done based on the scores received for both analysis.

In research of "Learn To Detect Phishing Scams Using Learning and Ensemble Methods" [13], Researcher has tried to recognize attackers' phishing emails from the legitimate mails by using several data mining approaches. Three data mining algorithms used to identify email scam types and their relation with phishing to avoid the users from receiving this kind of detected scams. Then used the collaborative methods considering all algorithms to improve the scam detection mechanism.

As described on "Lexical URL Analysis for Discriminating Phishing and Legitimate Email Messages" [14] study, a lexical URL analysis (LUA) technique has been introduced to improve the accuracy of classification for anti-phishing email filters. This technique has been mainly used to classify phishing websites. But it also proved to be more effective to classify emails too which contains URLs. Since currently most phishing emails contain with phishing URLs.

The researcher of "Phish Mail Guard: Phishing Mail Detection Technique by using Textual and URL Analysis" [15], has been proposed a hybrid method to detect phishing mail with combination of blacklist, white list and heuristic method. As the researcher found that most of the phishing mails have similarities in the text of email, therefore he has used textual analysis combined with URL analysis to get more effective outcome by referring to the previous studies.

"In Phish Catch – A Phishing Detection Tool" [16], researcher designed an algorithm to detect phishing after analyzing number of phishing attacks. This algorithm has been

focused warning the user on suspected link after identifying it as a phishing link and make a data warehouse containing with phishing information further. This data warehouse can be used to get information on the trends of phishing. Phish detection rate of this technique is less.

In the Research of "Phishing Detection Using Neural Network" [17], Researcher apply multilayer feed forward neutral networks to detect phishing emails and evaluate the effectiveness. A neural network (NN) systems has been implemented by the researcher after designing a feature set by processing the phishing data. Performance of the NN system has been evaluated using cross validation. Compared to the other major machine learning algorithms, researcher proved the satisfactory accuracy by doing a statistical analysis. Feature selection has been done by considering the phishing email characteristics. In this research, feed forward neural network has been used incorporate with some basic features related to email structure and links available to detect the phishing attacks.

"Phishing Email Detection Technique by using Hybrid Features" [18] research proposed a technique to detect phishing by considering hybrid features such as structure-based and behavior-based features including domain sender, blacklist words in subject and content, IP address in URL, dots in URL, symbols in URL, unique sender, unique domain, inconsistent hyperlink and return path. SVM has been use as the data mining algorithm in this technique. This researcher mentioned that the effectiveness of unique sender and unique domain features depend on the quality and size of corpus may happen time consuming. Also highlighted the unavailability of up to date blacklisted keyword list which has effected for the blacklist keywords feature. Also this research mentioned that Current implementation has not been considered the graphical features for the email such as images, banners and logos.

In paper [19], researcher has implemented Phish Storm: an automated phishing detection system to analyze whether any URL is linked with phishing site in real time. Experimental evidence has been used for this observation to detect phishing sites. In this paper, defined a new concept of intra-URL relatedness and evaluate by using features extracted from

words from Google and Yahoo search engines result data. These features are then used in machine learning approach classification to detect phishing URLs.

According to Research of "DC Scanner: Detecting Phishing Attack" [20], an email scanner (DC scanner) has been implemented which can be recognized the phishing URLs received with email messages for users. In this work, there are two steps to check on the details. As the first phase, check the html content of every link on email to verify domains of every link. As the second phase, script codes of the web pages have been checked for the malicious URL's. Also check whether phishers try to do modify the html tags and analyze the domains and related authority details of the links, script codes associated with web pages. This has given an idea to determine the probability of phishing attacks.

2.3 Summary of Related Studies

According above related studies, approaches have been categorized to four main categories based on the features they have used for phishing detection as shown on Table 2.1.

No	Research	Email Header Based	Email Body Content Based	URL - Lexical Based	URL Web Page Content Based
1	A Novel Algorithm to Detect Phishing URLs [4]				
2	A PageRank Based Detection Technique for Phishing Web Sites [5]			Google's Page Rank	
3	A Sender-Centric Approach to Detecting Phishing Emails [6]	only Sender			
4	Client-Side Counter Phishing Application using Adaptive Neuro-Fuzzy Inference System [7]				
5	Detecting Phishing Emails the Natural Language Way [8]				
6	Detecting Phishing Emails Using Hybrid Features [9]				
7	Detecting phishing e-mails using Text and Data mining [10]				

8	Detection of Phishing Emails using Feed Forward Neural Network [11]				
9	Identification and Detection of Phishing Emails Using Natural Language Processing Techniques [12]				
10	Learn To Detect Phishing Scams Using Learning and Ensemble Methods [13]				
11	Lexical URL Analysis for Discriminating Phishing and Legitimate E-Mail Messages [14]				
12	PhishMailGuard - Phishing Mail Detection Technique by using Textual and URL Analysis [15]				
13	PhishCatch – A Phishing Detection Tool [16]				
14	Phishing Detection Using Neural Network [17]				
15	Phishing Email Detection Technique by using Hybrid Features [18]				
16	PhishStorm - Detecting Phishing With Streaming Analytics [19]				
17	Research of the Anti-Phishing Technology Based on E-mail Extraction and Analysis [21]				
18	DC Scanner - Detecting Phishing Attack [20]			Only domain	Depend on web page design

Table 2.1 Comparison of related work

2.4 List of Features

After studying the previous work, Set of features has been selected and defined to capture the characteristics of phishing emails. Among the 30 selected features, there are some features defined based on observations. And these hybrid features have been used to implement the new classification model.

2.4.1 Email Header Based

The header consists of several pre-formatted content such as From, ReplyTo, Subject, Message Id etc.

F1 - ReplyTo domain is Not Equal to Sender domain: This feature indicate the comparison of sender domain and replyto domain.

F2 - Subject Content phishing word: Check whether email subject contained with pre-defined phishing key words which commonly found in phishing emails. Based on observations and literature review, phishing key words have been identified and listed such as account, confirm, offer, statement, urgent, verify etc.

F3 - Content Type: As per the MIME standard, Email could have multipart for Content-Type attribute with in the same email structure. Containing a multiple body part will be advantage to make a phishing email. This feature check that email has multipart content type or not.

2.4.2 Email Body Based

F4 - Addressing method of the recipient: Generally phishing email does not address to a particular recipient by name. It always starts with common addressing phrase such as “Hi Dear”, “Dear Customer”, “Dear Sri/Madam”, “Dear Friend”, “Dear Recipient” and “Dear Beneficiary” etc.

F5 - HTML Body: Email Body format could be HTML or Text or both. Most of the time phishers use to victimize users by HTML formatted emails. Checking email body format will be more important feature.

F6 - Has Script Code: This feature check the availability of any script code in email body such as java script.

F7 - Has Phishing words in email content: Check whether email body has contained with pre-defined phishing key words.

F8 - Has Form/Input tag: This feature define the availability of form tag or input tag in email body.

F9 - Has any URL link: Availability of URL has been check by this feature.

2.4.3 URL Based

Under this category, analyze the URLs of emails to get data for below features.

F10 - No. of images used as link: Number of URLs appears as images in the email content has been counted by this feature. These images can be used to make emails look legitimate by the phishers.

F11 - No. of domains: This feature represents the number of domain names of URLs which contain in email.

F12 - No. of deceptive links: This feature count the number of deceptive links within the email. Phishers use to mislead email users by these deceptive links which changing URL pointing location from the visible URL.

F13 - URL is a File: This feature check whether the email URL is locating a file such as word, pdf, exe file or locating a page of website.

F14 - Has shorten URL: This feature identify whether the URL has been shorten or not. Phishers use shortened URL to reduce the length and complexity of URL. There are many shortened URL service providers who are facilitating this free of charge such as goo.gl, tinyurl.com, adf.ly, bitly.com and swarife.com.

F15 - Has different domain: This feature check whether URL domain is similar to email domain.

F16 - Length of URL: This feature measure the length of URL.

F17 - Has IP in URL: This feature check the existence of IP address in any URL of the email.

F18 - Has Port in URL: It will check whether given URL contains port number or not.

F19 - Not Use SSL: URL is https or not. Normally https use for secure web sites.

F20 - No. of Slashes in URL: This feature measure the number of slashes in the URL. These slashes exhibit the availability of sub- folders which is used to hide the web page data.

F21 – No. of dots in URL: This feature checks the number of dots in a URL. Phishing sites will contain more number of dots.

F22 - No. of Dashes: This feature measure the number of dashes in the URLs.

F23 - No. of URLs have Phishing Words: Number of URLs which contain the phishing words will be measured by this feature.

F24 - URL Blacklist Status: URL is blacklisted or not will be checked by this feature. Phish Tank API is used to get the blacklist status of URLs.

F25 – URL Alexa Rank value: Alexa rank value is a most significant measure of web site traffic. This feature indicate the Alexa rank value of URLs.

2.4.4 URL Web Page Content Based

This category includes features which are extracted from content of web pages directed by URL of emails.

F26 - Has Form tag: This feature will be checked the availability of form tag on the web page.

F27 - Has Script: This feature will be checked the availability of script tags on the web page.

F28 - Has Input Control: Availability of input control will be indicated by this feature.

F29 - Has Iframe: Availability of Iframe control will be checked by this feature.

F30 - Has Phishing Words: Check whether web page contained with pre-defined phishing key words.

2.5 Summary

This chapter presented a complete literature review use of phishing detecting classification with a specific reference to data mining. We have defined the research problem and also identified the enhance email features addressing the research problem. We also identified the possible classification technique that can be used to address the research problem. Next chapter will discuss the technologies adapted for solving our problem.

Chapter 3

Technologies and Tools Used for Phishing Classification

3.1 Introduction

In the previous chapter we discussed previous works findings in the area of phishing detection, classifications and email features and we define our research problem and also identified features that we are going to evaluate using data mining classification as the technology to address the problem. This chapter highlights the effectiveness of selected technology that distinguishes it from the technologies applied in existing literature.

3.2 Data Mining Techniques

Data mining is a procedure which is used to find new information and predict future trends by identifying patterns & relationships in a large set of data. This has been done by various data analysis methods. There are several data mining techniques such as Association rules, Classification, Clustering, Predictions, Sequential Patterns, and Regression etc. Among these techniques, classification will be used to build phishing detecting model in this research. Classification use to assign new data to the relevant classes based on previous categorized data and help to do predictions for the future. Naive Bayes, k-Nearest Neighbors & Decision Tree will be used as classification algorithms with relevant to this research.

3.3 Naive Bayes

Naive Bayes is a classification algorithm which is based on the Bayes' theorem. According to that, there is an assumption of independence of a particular feature among the other features in a class. This classifier is a widely applied and simple statistical method which can be used for a large set of data to do classification considering the probability. It brings more efficient and accurate outcome than complex methods. Therefore, this is using most commonly in email filtering, recommendation systems and sentiment analysis etc.

3.4 k-Nearest Neighbors

The K- Nearest-Neighbors (k-NN) algorithm is a basic and simple non- parametric method which can be used for regression as well as for the classification. There are two steps of applying this algorithm. As first step, need to train the algorithm to recognize certain classes. Then, it can be used to make an educated guess based on the classification. KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label of classification method or averages the labels of regression method.

3.5 Decision Tree

Decision Tree is one of the most powerful and popular method for both prediction and classification. It is a flowchart which is very simple to understand the data and very comfortable with human level thinking to make some good interpretations. It has a structure like a tree with internal nodes, branches and leaves. Each node represents a feature or attribute and branch hold a class label or decision rule while each leaf represents the outcome. It is called as root node for the topmost node in a decision tree. The complexity of a decision tree depends on the number of features and number of records of the relevant data set. It is non- parametric and distribution-free method which can be used for high dimensional data to get more accurate results. Using feature selection measures, each feature which related to the given dataset has been ranked with scores. Most popular selection measures are Information Gain, Gain Ratio, and Gini Index. Best scored feature can be selected to split the dataset in to possible fine way. This best feature is considered as a decision node and divide the data set in to subsets. By repeating the same process, decision tree can be built up.

3.6 Rapid Miner studio

Rapid Miner studio is a powerful and productive data mining software which can be used for vast quantities of data. It provides platform with graphical user interface that functions as an integrated environment for machine learning, data preparation, deep learning,

predictive analytics, and text mining. In this research. This software has been used for the data mining process to get more accurate outcome.

3.7 .Net Framework

In this research, an email classification tool for detecting phishing has been developed as the solution for the research problem using C# .net framework 4.5 programing language developed by Microsoft. There are two parts, one is feature data extraction tool and final solution is classification tool. Data extraction tool was developed by using .Net windows platform as a console application. The main classification tool was developed with asp.Net platform as a web system.

3.8 Microsoft Visual Studio

Microsoft Visual Studio is software product which has been developed and provided by Microsoft Corporation. It is used as an Integrated Development Environment tool for developing windows applications, web applications, mobile applications as well as various open source applications. It mainly provides environment for .Net formwork applications, but recently they have provided facilities to open source platforms too. In this research we have planned to use this as an IDE tool to develop the system.

3.9 EAGetMail

EAGetMail [24] is a .Net Nuget package library which was developed by Ivan Admin System. This library can be used to retrieve emails from POP3 and IMAP4 email servers and parsing email components. In this research, it has been used for .EML files reading process in the extraction tool and also used for retrieving email from email account in the classification tool.

3.10 HtmlAgilityPack

HtmlAgilityPack [23] is an agile HTML parser that builds a read/write DOM. It is freely available on .Net Nuget packages library which is published by ZZZ Project Team. This library has been used as a HTML parser to extracting data of features from email HTML content.

3.11 TAlex.SEOStats

TAlex.SEOStats [25] is a .Net Nuget package library which is owned by Alex Titarenko. This library has been used for Alexa Ranking value capturing process. In this research Alexa rank has been considered as a key feature of classification model. In data extracting tool used this for collecting data for Alexa rank value.

3.12 PhishTank API

PhishTank [26] is a free community site which is collaborative clearing house for data and information about phishing on the Internet. It is operated by OpenDNS, a company founded in 2005. Anyone can use the PhishTank freely with relevant to phishing data by submitting, verifying, tracking and sharing. This site provides open API for researchers and developers to integrate their data base of phishing. With reference to the blacklisted database, any URL can be checked whether it has been included in that list as a phishing site. This has been considered for F24 feature in feature extraction process of this research.

3.13 Microsoft SQL Server

Microsoft SQL Server is a relational database management system and leading database technology which has been developed by Microsoft Corporation. It is a full featured database software product and It has own query language which can facilitate sorting and retrieving data. Not only that, it has separate components for data analysis and intelligence reporting. This Product is working as a client- server architecture and it provides services to other applications, hosted on same server or on another computers via network. In this research, MS SQL Express 2017 version has been used as the database and SQL has been used as the query language.

3.14 Summary

This chapter describes the technologies and tools which have been used to complete this research. In here, Data mining is used as the main approach to implement the classification data model. The next chapter shows a novel approach to classify phishing emails by using the presented technologies and tools in this chapter.

Chapter 4

A Novel Approach of Classification Phishing Email using Hybrid Features

4.1 Introduction

Chapter three presented the technology to be used to solve the research problem. This chapter described our approach to address the problem of detecting phishing email accurately by using a web-based system. We present our approach by highlighting hypothesis, input, output, process, users and features of the solution.

4.2 Hypothesis

As per the hypothesis, the problem of detecting phishing mails can be solved by using the proposed classification tool. Per the hypothesis the accuracy of tool will be high due to hybrid features considering email header data, email content, URL analysis & URL page content. We are going to use various classify technologies such as Naïve Bayes, KNN & Decision Tree. Then finally picked up most accuracy classifying techniques based on data model.

4.3 Input

We are using .EML format files as the input for extraction of features data. EML is a universal file extension of email messages which was developed by Microsoft Corporation. It is a supported file extension for Microsoft outlook email client as well as other email clients of other service providers. EML file saves email message as a plan text with specific format. It is contained header details with sender, recipient, domain and main message as well as HTML tags. For this research, phishing email .EML files have been downloaded from Malware-Traffic-Analysis.net [22] phishing blog and legitimate email .EML files have been collected from known email accounts.

4.4 Output

As main output of this research, provides a web based email classification tool that can detected phishing emails. It included main components as below,

-Classification model based on hybrid features

-Web based system to classify emails whether phishing or legitimate.

4.5 Process and Features

Using .EML file, extract the features data by extraction tool which is developed by .Net framework. Data Collection and Pre- processing of data have been done by this tool. Collected data is used to data mining process to build a classification model. This model is created by using rapid miner software. Then compare the accuracy by applying several classification techniques and evaluate the model. After that, based on evaluated classification model, we build the web based system to classify emails. The system has the facility to retrieve emails from email accounts and detect phishing emails.

4.6 Users

Every organization employee who is using emails in day today tasks as regular communication platform or any individual who has email accounts.

4.7 Summary

This chapter describes our novel approach to classify emails to detect phishing emails. It pointed out how this research offers an efficient and accurate solution for web-based system using data mining. The next chapter shows the design of the presented solution.

Chapter 5

Design of the Classification Tool

5.1 Introduction

Chapter four presented the approach to develop the Email Classification tool by using data mining techniques. This chapter describes the overall picture of the proposed solution design.

5.2 High level Architecture of System

In this research we have developed two main tools. One is console application for data extraction to collect data set for building the classification mode. And other one is proposed solution system. It is web application which is developed based on evaluated classification model. High level architecture of this system is illustrated by Figure 5.1. Features extraction tool and classification tool are used same service layer and data access layer with common Data base. Whole system design on layered architecture since it will be easier to maintain.

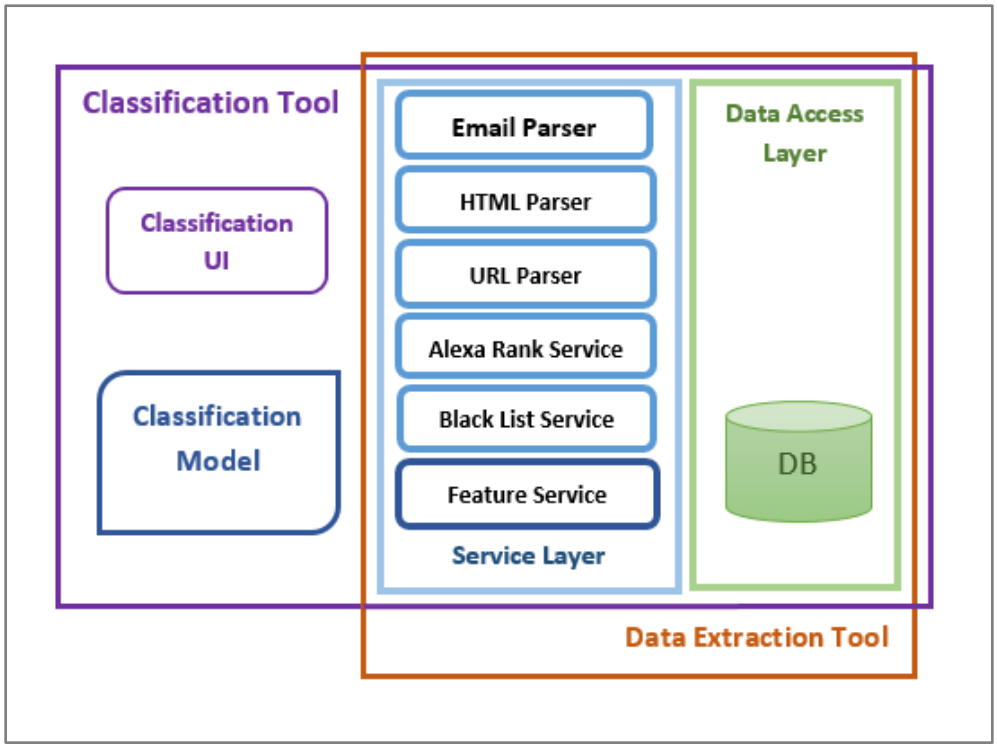


Figure 5.1 High Level Architecture of System

5.3 Data Collection and Preprocessing

Data collecting and preprocessing process have been done by developed feature extraction tool. As input data, we have collected .EML files for feature extraction. Phishing email files have been downloaded from Malware-Traffic-Analysis.net [22] and legitimate emails was downloaded by know email accounts. Extraction tool is designed to read .EML files and extract email componet by using email parser. That process was done by EAGetMail email parser. HTML parser and URL parsers design for extracting html componet by email content. HtmlAgilityPack Nuget library helps to do HTML and URL parsing process.

Feature Service process do the important part of data collection and preprocessing step. It is extracted data related to features which was focus to do classification and preprocess. Some data can be dereclty retrived from parsers but some features cannot. Alexa rank service was used for get alexa rank feature by Talex.SEOStats nuget library. Black list service was designed for get black list sataus of URLs via PhishTank API. As mention above chapters, 30 features related data can be extracted by feature service process. Finaly extraction tool was facilitated to download data as excel file acording to selected 30 hybrid features. That file can be directly used for data mining classification process using Rapid Minner software.

5.4 Design of Classification Tool

Figure 5.2 illustrate the high level process diagram of proposed classification tool. Tool has the facility to retrieve email box with given credential related account. Then email parser, HTML parser and URL parser done their job. After that feature service does the extraction process of relevant features data by applying the features data set to build classification model to detect phishing emails. This tool has the facility to see which mails are phishing and which mails are legitimate. Phishing emails are highlighted with red color.

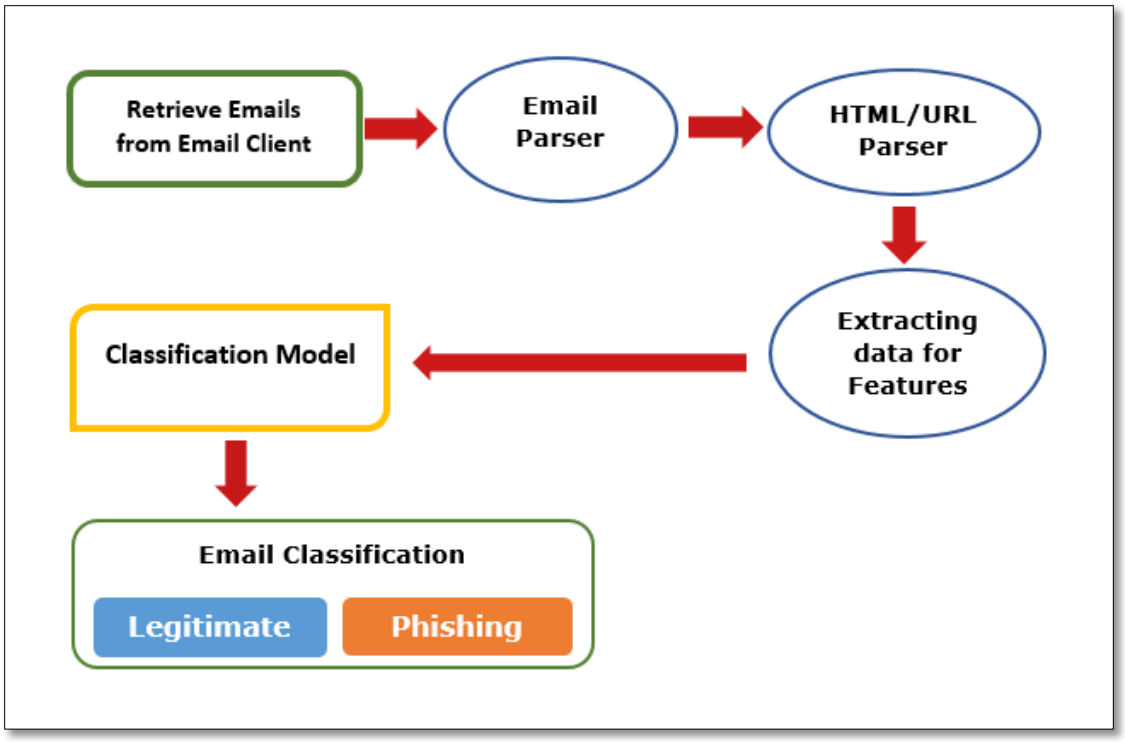


Figure 5.2 High Level Process Diagram of Classification Tool

5.5 Back end Database

The back-end database is the key to handle input and output data to each and every service of provided solution. This is also an essential component to work with the User interface. The design of this database is rather critical activity in the solution. This affects the classification feature set as well as user interface. The database consists all parsing data and hybrid features data of every emails and other supported tables to minimize data redundancy. According to this design without going through classification the system facilitates sessions to query on the database for ad-hock functioning. More importantly, interaction between system and back-end database through data access layer only. It makes increase reusability and efficiency of the whole system.

5.6 Summary

This chapter discussed the design of Classification email solution. As mentioned earlier, we designed the solution with separate layers as User Interface, Service layer and Data layer with Back-End Database together with their roles. In next chapter will be described the implementation of this solution.

Chapter 6

Implementation of Classification Tool

6.1 Introduction

In previous chapter, we have mentioned the overall design of the proposed solution. This chapter will describe implementation details of each of the sections mentioned in the previous chapter.

6.2 Core Service

Core functions of this solution are wrapped as service layer for better usability. UI layers of Extraction and Classification tool will communicate with the service layer for core functions. Email and HTML parser are integrated with service layer. Feature service extracts parsers data to get feature's data. F25 and F26 features get data from external sources. PhishTank API integrated with get blacklist status of URLs included in email contents for F25 feature. It is a rest API, that communicates by parsing JOSN result. TAlex.SEOStats Nuget Library intergarted for get Alexa Rank value of URLs to F26 feature.

6.3 Data Collection by Data Extraction Tool

After referring to available literature and observations of phishing emails, 30 hybrid features which captures the characteristics of phishing emails have been selected for conducting the data mining process. To extract the relevant data for these features we have to develop separate tool for that. These hybrid features cannot be directly extract from email. The data related to features are content email header, email body, URL of email content and Content of URL page. This process needs special techniques and processes for itself. We developed .Net console application reference to our core service layer. Our data source is .EML files which was downloaded on Malware-Traffic-Analysis.net [22] for phishing emails and legitimate emails files form known email accounts. 471 phishing emails and 407 legitimate emails, Al together collect 878 phishing and legitimate email

files. First step of extraction tool was collected email header, body data from collected .EML files. Email parser can be used for accommodate this process. Next step is to extract the HTML and URL data by HTML parser. Thereafter, we can process data extraction phase for pre-defined features. The final output of extraction tool is Excel file of features data set.

6.4 Classification Model

As stated, Chapter 4, Rapid Miner Studio is the software we have used for Data mining tool. Extracted and preprocessed features data by extraction tool can be applied to data source for rapid miner. We have used Naive Bayes, K-NN and Decision Tree as classification techniques. Based on those accuracy we can evaluate the model. We have split data set for training and testing as 70% and 30%. According to experimental result, this email classification model has been recognized. By using the model, relevant and optimized features have been selected and used for the implementation of final email classification tool for phishing detection.

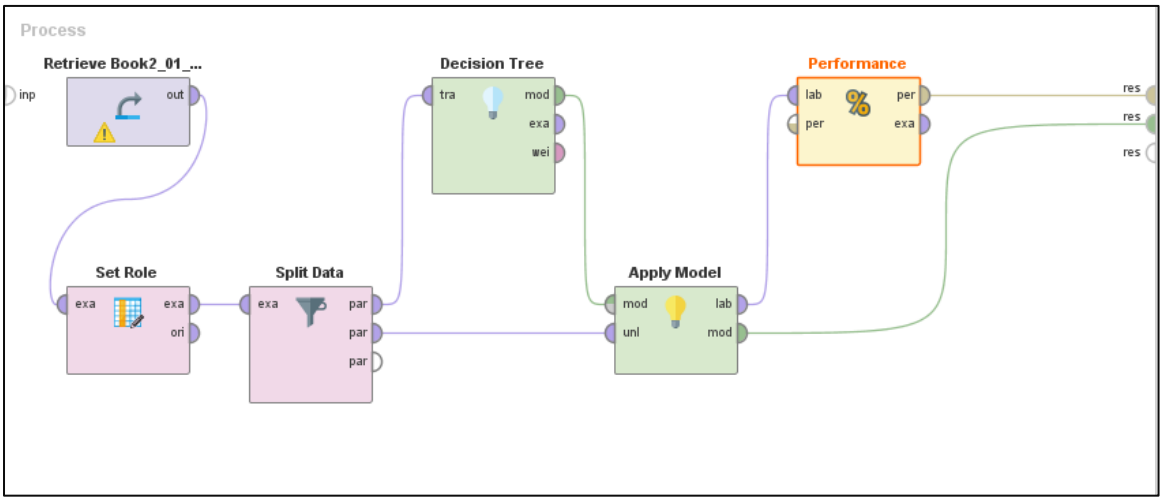


Figure 6.1 Decision Tree Models using Rapid Miner

6.5 Classification Tool

Final classification tool is a web-based system which is implemented as .Net based application running on any web browser. The implementation of tool is been based on ASP.net MVC architecture and communicate with service layer for relevant functions. As

described in chapter 5, Data Access Layer has been developed by integrated with MS SQL Express to store and retrieve data. System has the facility to read email from authorized email accounts or .EML files as an input source of prediction. Classification process of the system has been done by the evaluated classification model. Extracted data of selected hybrid features has provided to the model to predict the input emails whether phishing or not. Selected features include 19 features which are related to header, email content, URL and Content of URL web page. Evaluated model used only selected significant features for accuracy and high performance. Figure 6.1 display the classification interface of the system using .EML files. More interfaces have been attached to Appendix.

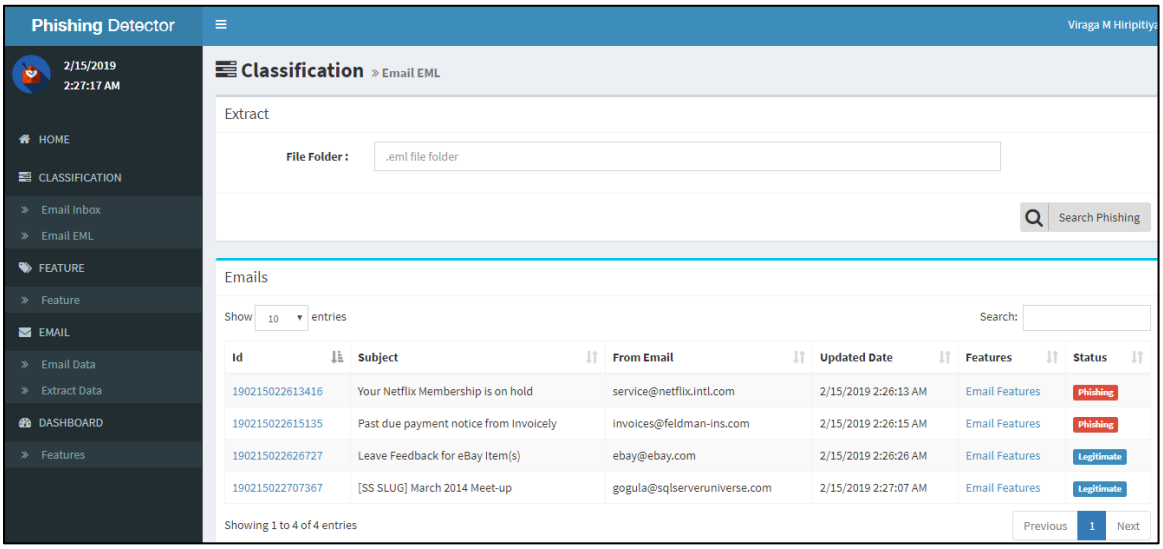


Figure 6.2 Classification UI of System

6.6 Summary

This chapter provides an overall implementation details of each module of the proposed solution. Moreover, it mentioned software and data mining techniques for models development with align to design. In the next chapter, we will evaluate all the modules implemented in the solution.

Chapter 7

Evaluation

7.1 Introduction

The previous chapter discussed the details of implementation of all the modules mentioned in the proposed solution. This chapter justifies and evaluates the overall solution, data mining techniques and data models used in Classification tool.

7.2 Evaluation of Classification Techniques

We have trained collected data set using different classification techniques namely Naïve Bayes, k-NN and Decision Tree by the help of Rapid Miner tool. Phishing detection prediction model is the main component of this classification system. Accuracy of the prediction will depend on the model. For evaluating a classifier efficient and performance we can use confusion matrix which can be evaluated various measurements such as accuracy, recall and precision.

- Accuracy – Percentage of all correct decisions this measures the percentage of all decisions that were correct.
- Recall - Portion of the completeness of correct categories that were assigned.
- Precision - Fraction of correctness.

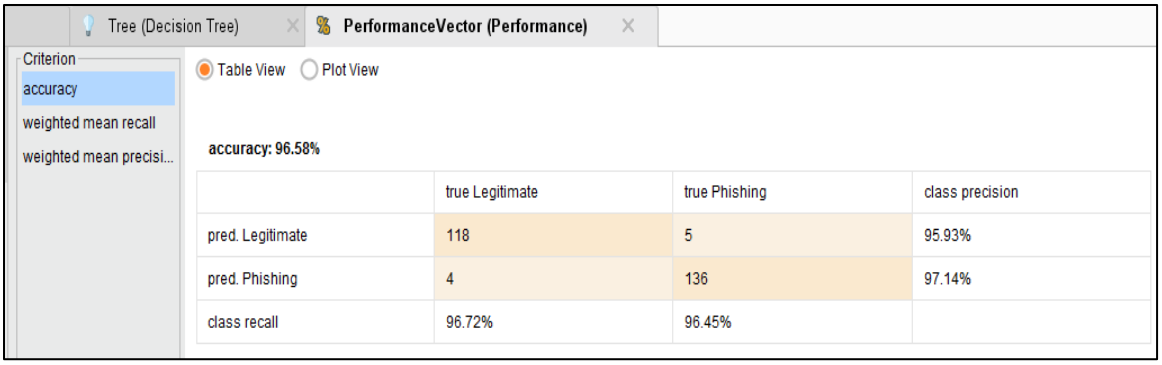
These measurements and their definition are given below Table 7.1

Technique	Accuracy	Recall	Precision
Naïve Bayes	82.89%	82.00%	84.82%
k-NN	95.06%	94.73%	95.60%
Decision Tree	96.58%	96.59%	96.54%

Table 7.1 Classification Performance

According to Table 7.2 shows that Naïve Bayes and k-NN represent low accuracy, but Decision Tree having higher accuracy value than other techniques. It was found that Decision Tree produced the best results in prediction of phishing emails. Other classification techniques as shown in above table did not perform significantly well in our

research. Figure 7.1 shows the accuracy of Decision Tree classification model. In our solution, we have applied DT classification model based on evaluated result. Figure 7.2 show the Decision Tree structure. Appendix C include the decision tree rules with selected features of the classification model. Classification tool is predict the final out put based on tree structure rules.



	true Legitimate	true Phishing	class precision
pred. Legitimate	118	5	95.93%
pred. Phishing	4	136	97.14%
class recall	96.72%	96.45%	

Figure 7.1 Accuracy of Decision Tree

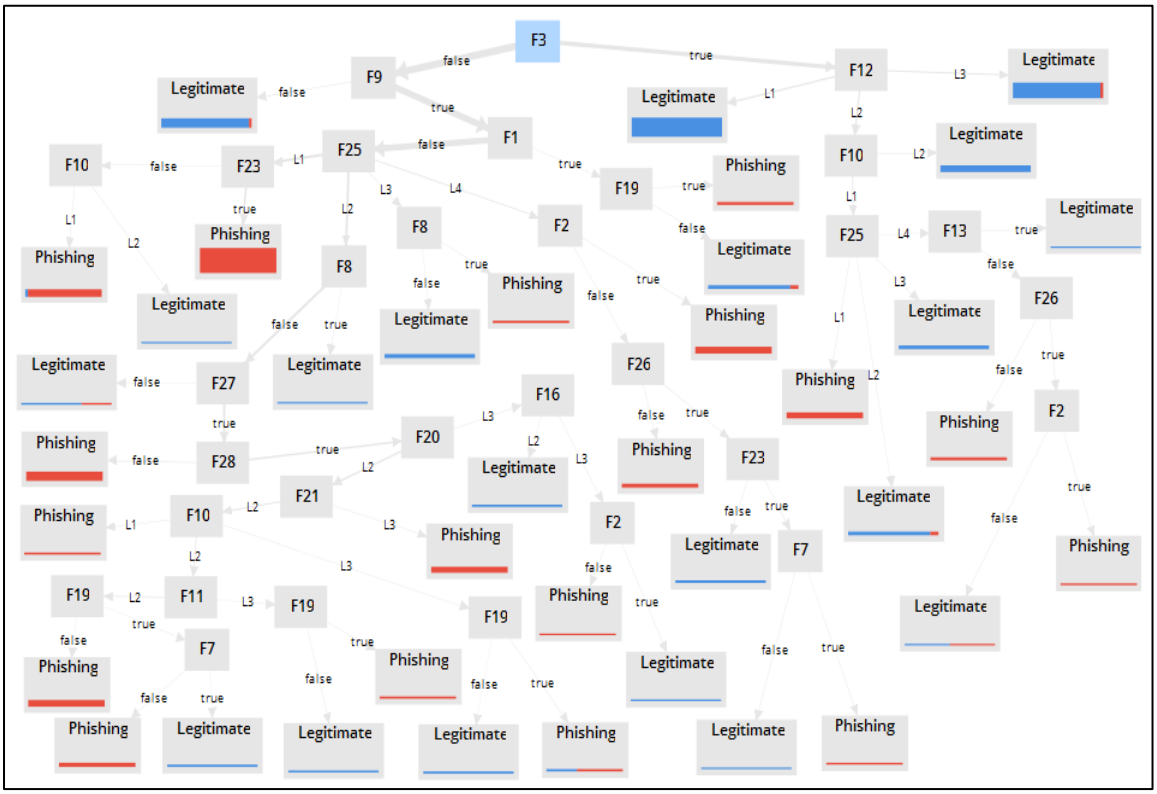


Figure 7.2 Decision Tree Structure

7.3 Summary

This chapter evaluated the methodologies and the results discussed in the implementation chapter. In the next chapter, we will discuss limitations and future improvements of Classification tool.

Chapter 8

Conclusion and Further Work

8.1 Introduction

In this research, we have developed email classification tool based on hybrid features. This phishing email detection technique will be very effective because of the possibility of reading phishing mails by user is very low. Accuracy of the tool will be very high because it covers all areas of phishing indicators like email content, URL analysis, content of URL pages , blacklist details and search engine result .

8.2 Limitations

Provided tool is not capable to read emails of user accounts automatically. Once emails received to user mail box, User will have to check manually phishing emails with our system. It will not automatically detect phishing emails.

User need to provide credential of email account to operate the Classification Tool. It may be a privacy concern for the user.

This tool will operate separately from email client application. Using two applications will not be convenient to the user.

8.3 Future Developments

With regard to the above limitation in the system, we can enhance the system to directly access email server with proper authorization, then classify the emails in top level. This will be the best solution for organizations to be used as an anti-phishing technique. Also we can develop email Add-ins for email clients like Microsoft Outlook. By developing outlook add-in based on our classification model, any outlook user can add this feature to their account. This add-in has the facility to classify the email automatically when phishing type emails are received. Providing that, Email credential entering requirement also can be avoided.

8.4 Summary

This final chapter concludes the thesis by describing the solution given with data mining to classify the phishing emails and address the limitation of proposed solution.

References

[1] J. McCabe, "FBI Warns of Dramatic Increase," 4 April 2016.

[2] APWG, "Phishing Activity Trends Report," 2017.

[3] J. Hong, "The Current State of Phishing Attacks," Carnegie Mellon University, 2012.

[4] V. R. Hawanna , V. Y. Kulkarni and R. A. Rane, "A Novel Algorithm to Detect Phishing URLs," International Institute of Information Technology, Pune, 2016.

[5] A.Naga Venkata Sunil and A. Sardana, "A PageRank Based Detection Technique for Phishing," IEEE Symposium on Computers & Informatics, 2012.

[6] F. Sanchez and Z. Duan, "A Sender-Centric Approach to Detecting Phishing," International Conference on Cyber Security, 2012.

[7] S. Singh, A. K. Sarje and M. Misra, "Client-Side Counter Phishing Application using Adaptive Neuro-Fuzzy Inference System," Fourth International Conference on Computational Intelligence and Communication Networks, 2012.

[8] R. Verma, N. Shashidhar and N. Hossain, "Detecting Phishing Emails the Natural Language Way," Springer-Verlag Berlin Heidelberg , 2012.

[9] L. Ma, B. Ofoghi, P. Watters and S. Brown, "Detecting Phishing Emails Using Hybrid Features," IEEE Computer Society, 2009.

[10] M. Pandey and V. Ravi, "Detecting phishing e-mails using Text and Data mining," 2012 IEEE International Conference on Computational Intelligence and Computing Research, 2012.

[11] N. G. M. Jameel and L. E. George, "Detection of Phishing Emails using Feed Forward Neural Network," International Journal of Computer Applications, 2013.

[12] S. Aggarwal, V. Kumar and S D Sudarsan, "Identification and Detection of Phishing Emails Using Natural Language Processing Techniques," https://www.researchgate.net/publication/288379774, 2014.

[13] A. Saberi, M. Vahidi and B. M. Bid, "Learn To Detect Phishing Scams Using Learning and Ensemble Methods," 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops, 2007.

[14] M. Khonji, Y. Iraqi and A. Jones, "Lexical URL Analysis for Discriminating Phishing and Legitimate E-Mail Messages," 6th International Conference on Internet

Technology and Secured Transactions, 11-14 December 2011, Abu Dhabi, United Arab Emirates, 2011.

[15] J. Hajgude and L. Ragha, "Phish Mail Guard :Phishing Mail Detection Technique by using Textual and URL Analysis," IEEE, 2012.

[16] W. D. Yu, S. Nargundkar and N. Tiruthani, "PhishCatch – A Phishing Detection Tool," Annual IEEE International Computer Software and Applications Conference, 2009.

[17] N. Zhang and Y. Yuan, "Phishing Detection Using Neural Network".

[18] L. . M. Form, K. L. Chiewy, . S. N. Szez and W. K. Tiong, "Phishing Email Detection Technique by using Hybrid Features".

[19] S. Marchal, J. François, . R. State and T. Engel, "PhishStorm: Detecting Phishing With Streaming Analytics," IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, 2014.

[20] B. Kumar, P. Kumar, . A. Mundra and S. Kabra, "DC Scanner: Detecting Phishing Attack," Third International Conference on Image Infonnation Processing, 2015.

[21] Y. Du and F. Xue, "Research of the Anti-Phishing Technology Based on E-mail Extraction and Analysis," International Conference on Information Science and Cloud Computing Companion, 2013.

[22] MalwareTraffic, "Malware-Traffic-Analysis.net," [Online]. Available: http://www.malware-traffic-analysis.net/2019/index.html.

[23] ZZZProjects, "Nuget Gallery - HtmlAgilityPack," zzz Projects, [Online]. Available: https://www.nuget.org/packages/HtmlAgilityPack.

[24] Ivan.AdminSystem, "NuGet Gallery - EAGetMail," Ivan.AdminSystem, [Online]. Available: https://www.nuget.org/packages/EAGetMail/.

[25] T. Alex, "Nuget Gallery - TAlex.SEOStats," T-Alex, [Online]. Available: https://www.nuget.org/packages/TAlex.SEOStats/.

[26] phishtank, "PhishTank - Join the fight against phishing," PhishTank, [Online]. Available: https://www.phishtank.com.

Appendix A - Sample .EML File

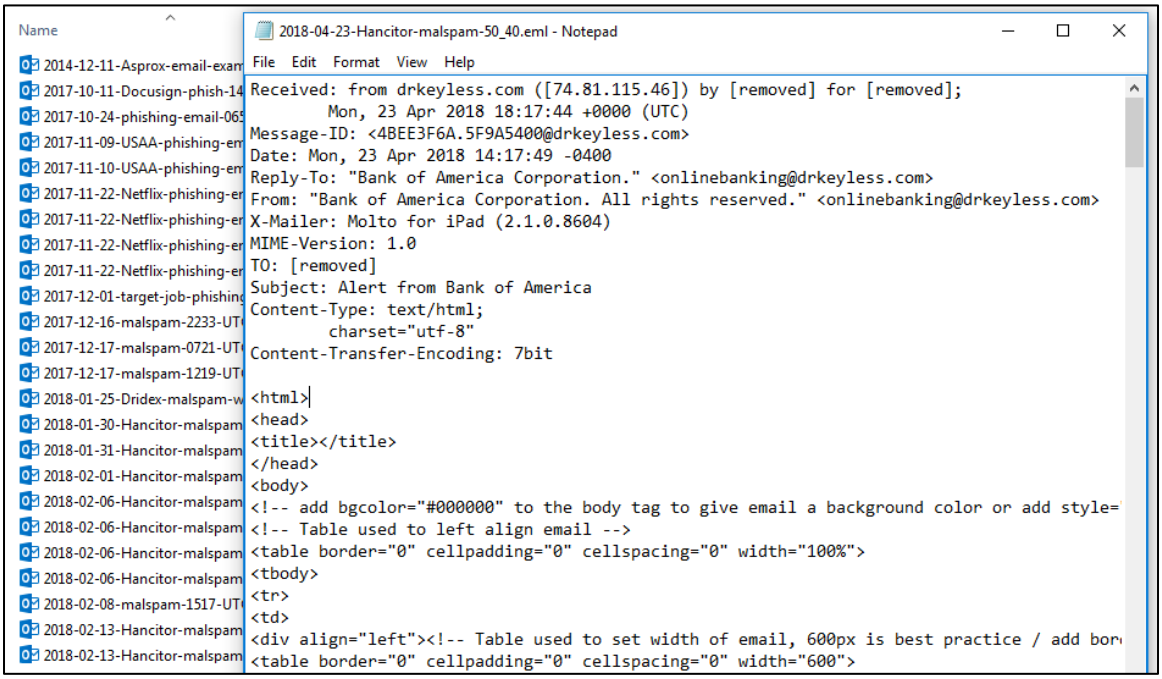


Figure A.1 .EML File

Appendix B - Model Evaluation Summary

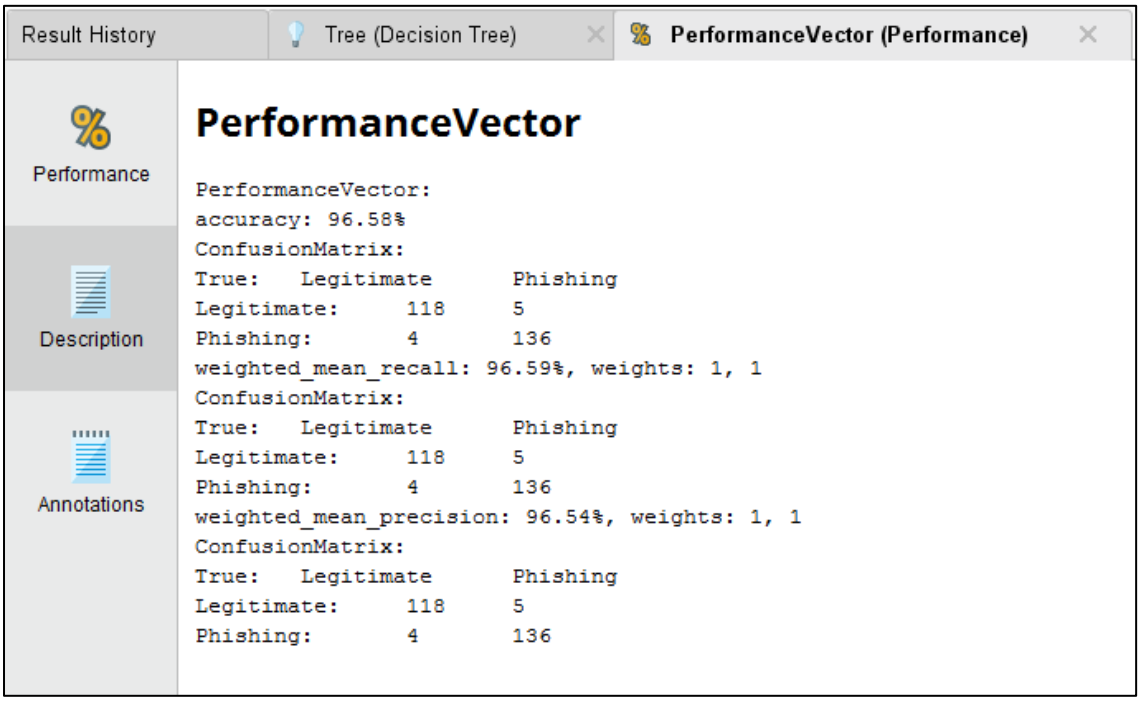


Figure B.1 Decision Tree Performance

Appendix C - Decision Tree Rules

```
F3 = false
|   F9 = false: Legitimate {Legitimate=35, Phishing=1}
|   F9 = true
|   |   F1 = false
|   |   |   F25 = L1
|   |   |   |   F23 = false
|   |   |   |   |   F10 = L1: Phishing {Legitimate=1, Phishing=27}
|   |   |   |   |   F10 = L2: Legitimate {Legitimate=2, Phishing=0}
|   |   |   |   F23 = true: Phishing {Legitimate=0, Phishing=104}
|   |   |   F25 = L2
|   |   |   |   F8 = false
|   |   |   |   |   F27 = false: Legitimate {Legitimate=2, Phishing=1}
|   |   |   |   |   F27 = true
|   |   |   |   |   |   F28 = false: Phishing {Legitimate=0, Phishing=35}
|   |   |   |   |   |   F28 = true
|   |   |   |   |   |   |   F20 = L2
|   |   |   |   |   |   |   |   F21 = L2
|   |   |   |   |   |   |   |   |   F10 = L1: Phishing {Legitimate=0, Phishing=4}
|   |   |   |   |   |   |   |   |   F10 = L2
|   |   |   |   |   |   |   |   |   |   F11 = L2
|   |   |   |   |   |   |   |   |   |   |   F19 = false: Phishing {Legitimate=0, Phishing=24}
|   |   |   |   |   |   |   |   |   |   |   F19 = true
|   |   |   |   |   |   |   |   |   |   |   |   F7 = false: Phishing {Legitimate=0, Phishing=13}
|   |   |   |   |   |   |   |   |   |   |   |   F7 = true: Legitimate {Legitimate=5, Phishing=0}
|   |   |   |   |   |   |   |   |   |   F11 = L3
|   |   |   |   |   |   |   |   |   |   |   F19 = false: Legitimate {Legitimate=4, Phishing=0}
|   |   |   |   |   |   |   |   |   |   |   F19 = true: Phishing {Legitimate=0, Phishing=5}
|   |   |   |   |   |   |   |   |   F10 = L3
|   |   |   |   |   |   |   |   |   |   F19 = false: Legitimate {Legitimate=5, Phishing=0}
|   |   |   |   |   |   |   |   |   |   F19 = true: Phishing {Legitimate=2, Phishing=3}
|   |   |   |   |   |   |   |   F21 = L3: Phishing {Legitimate=0, Phishing=21}
|   |   |   |   |   |   |   F20 = L3
|   |   |   |   |   |   |   |   F16 = L2: Legitimate {Legitimate=5, Phishing=0}
|   |   |   |   |   |   |   |   F16 = L3
|   |   |   |   |   |   |   |   |   F2 = false: Phishing {Legitimate=0, Phishing=2}
|   |   |   |   |   |   |   |   |   F2 = true: Legitimate {Legitimate=2, Phishing=0}
|   |   |   |   F8 = true: Legitimate {Legitimate=2, Phishing=0}
|   |   |   F25 = L3
|   |   |   |   F8 = false: Legitimate {Legitimate=13, Phishing=0}
|   |   |   |   F8 = true: Phishing {Legitimate=0, Phishing=5}
|   |   |   F25 = L4
|   |   |   |   F2 = false
|   |   |   |   |   F26 = false: Phishing {Legitimate=0, Phishing=12}
|   |   |   |   |   F26 = true
|   |   |   |   |   |   F23 = false: Legitimate {Legitimate=7, Phishing=0}
|   |   |   |   |   |   F23 = true
|   |   |   |   |   |   |   F7 = false: Legitimate {Legitimate=2, Phishing=0}
|   |   |   |   |   |   |   F7 = true: Phishing {Legitimate=0, Phishing=3}
|   |   |   |   F2 = true: Phishing {Legitimate=0, Phishing=25}
|   |   F1 = true
|   |   |   F19 = false: Legitimate {Legitimate=10, Phishing=1}
|   |   |   F19 = true: Phishing {Legitimate=0, Phishing=8}
F3 = true
|   F12 = L1: Legitimate {Legitimate=79, Phishing=0}
|   F12 = L2
|   |   F10 = L1
|   |   |   F25 = L1: Phishing {Legitimate=0, Phishing=21}
|   |   |   F25 = L2: Legitimate {Legitimate=10, Phishing=1}
|   |   |   F25 = L3: Legitimate {Legitimate=11, Phishing=0}
|   |   |   F25 = L4
|   |   |   |   F13 = false
|   |   |   |   |   F26 = false: Phishing {Legitimate=0, Phishing=8}
|   |   |   |   |   F26 = true
|   |   |   |   |   |   F2 = false: Legitimate {Legitimate=1, Phishing=1}
|   |   |   |   |   |   F2 = true: Phishing {Legitimate=0, Phishing=3}
|   |   |   |   F13 = true: Legitimate {Legitimate=2, Phishing=0}
|   |   F10 = L2: Legitimate {Legitimate=24, Phishing=0}
|   F12 = L3: Legitimate {Legitimate=61, Phishing=2}
```

Figure C.1 Decision Tree rules

Appendix D – Code Snippet

```
public EmailItem ParseEmail(string emlFile)
{
    EmailItem item = new EmailItem();
    Mail oMail = new Mail("TryIt");
    oMail.Load(emlFile, false);
    DateTime now = DateTime.Now;
    item.EmailItemId = Convert.ToInt64(now.ToString("yyMMddHHmmssfff"));
    item.FileName = emlFile;
    MailAddress fromMail = oMail.From;
    item.FromEmail = fromMail.Address;
    item.FromName = fromMail.Name;
    item.FromDomain = fromMail.GetAddressDomain();
    MailAddress[] addrs = oMail.To;
    int maxTo = addrs.Length;
    if (maxTo > 10)
        maxTo = 10;
    for (int i = 0; i < maxTo; i++)...
    item.EmailSubject = oMail.Subject.Replace("(Trial Version)", "");
    item.BodyText = oMail.TextBody;
    item.BodyHTML = oMail.HtmlBody;
    item.IsEncrypted = oMail.IsEncrypted;
    item.IsReport = oMail.IsReport;
    item.IsSigned = oMail.IsSigned;
    item.OriginalBodyFormat = oMail.OriginalBodyFormat.ToString();
    item.MessageClass = oMail.MessageClass;
    MailAddress replyMail = oMail.ReplyTo;
    item.ReplyToEmail = replyMail.Address;
    item.ReplyToName = replyMail.Name;
    item.ReplyToDomain = replyMail.GetAddressDomain();
    item.ReceivedDate = oMail.ReceivedDate;
    item.SentDate = oMail.SentDate;
    Collection<EmailReceivedLog> rLogs = new Collection<EmailReceivedLog>();
    HeaderCollection oHeaders = oMail.Headers;
    int count = oHeaders.Count;
    for (int i = 0; i < count; i++)
    {
        HeaderItem oHeader = oHeaders[i] as HeaderItem;
        if (oHeader.HeaderKey == "MIME-Version")
            item.MIMEVersion = oHeader.HeaderValue;
        else if (oHeader.HeaderKey == "Date")
            item.HDate = oHeader.HeaderValue;
        else if (oHeader.HeaderKey == "X-Priority")
            item.XPriority = oHeader.HeaderValue;
        else if (oHeader.HeaderKey == "X-Mailer")
            item.XMailer = oHeader.HeaderValue;
        else if (oHeader.HeaderKey == "Message-ID")
            item.MessageId = oHeader.HeaderValue;
        if (oHeader.HeaderKey == "Content-Type")
            item.ContentType = oHeader.HeaderValue;
```

Figure D.1 Email Parser

```
4 references | 0 exceptions
public Collection<EmailURL> GetEmailURLs(string html, long emailItemId)
{
    Collection<EmailURL> items = new Collection<EmailURL>();
    var htmlDoc = new HtmlDocument();
    htmlDoc.LoadHtml(html);
    var nodes = htmlDoc.DocumentNode.SelectNodes("//a");
    string innerHtml = "";
    if (nodes != null)
    {
        foreach (var n in nodes)
        {
            EmailURL item = new EmailURL();
            item.EmailItemId = emailItemId;
            item.Type = "URL";
            item.HTMLTag = n.OuterHtml;
            item.Address = n.Attributes["href"].Value;
            innerHtml = n.InnerHtml;
            if (item.Address != null && item.Address != "")
            {
                item.URLText = n.InnerText;
                UrlDTO url = new UrlDTO();
                url = GetUrlData(item.Address);
                item.URLDomain = url.UrlDomain;
                item.URLPort = url.UrlPort;
                item.URLProtocol = url.UrlProtocol;
                item.IsIP = url.IsIP;
                item.IsSSL = url.IsSSL;
                item.IsPort = url.IsPort;
                item.IsShorten = url.IsShorten;
                item.PageContentHTML = GetUrlPageContentHTML(item.Address);
                item.PageContentText = GetUrlPageContentText(item.Address);
                PhishTankService _phishTankService = new PhishTankService();
                item.IsBlacklisted = _phishTankService.IsBlacklisted(item.Address);
                AlexaRankService _alexaRankService = new AlexaRankService();
                item.AlexaRank = _alexaRankService.GetAlexaRank(item.URLDomain);
                item.IsForm = IsIncludedForm(item.PageContentHTML);
                item.IsScript = IsIncludedScript(item.PageContentHTML);
                item.IsInput = IsIncludedInput(item.PageContentHTML);
                item.IsIframe = IsIncludedIframe(item.PageContentHTML);
                if (item.PageContentText != "")
                    item.IsPhishingW = IsIncludedPhishingWord(item.PageContentText);
                item.IsImageLink = false;
                if (innerHtml.ToLower().Contains("img"))
                    item.IsImageLink = true;
                if (item.URLPort.Trim() != "25" && item.URLPort.Trim() != "0" && item.URLPort.Trim() != "-1")
                    items.Add(item);
            }
        }
    }
    return items;
```

Figure D.2 HTML Parser

Appendix E – Classification Tool UI

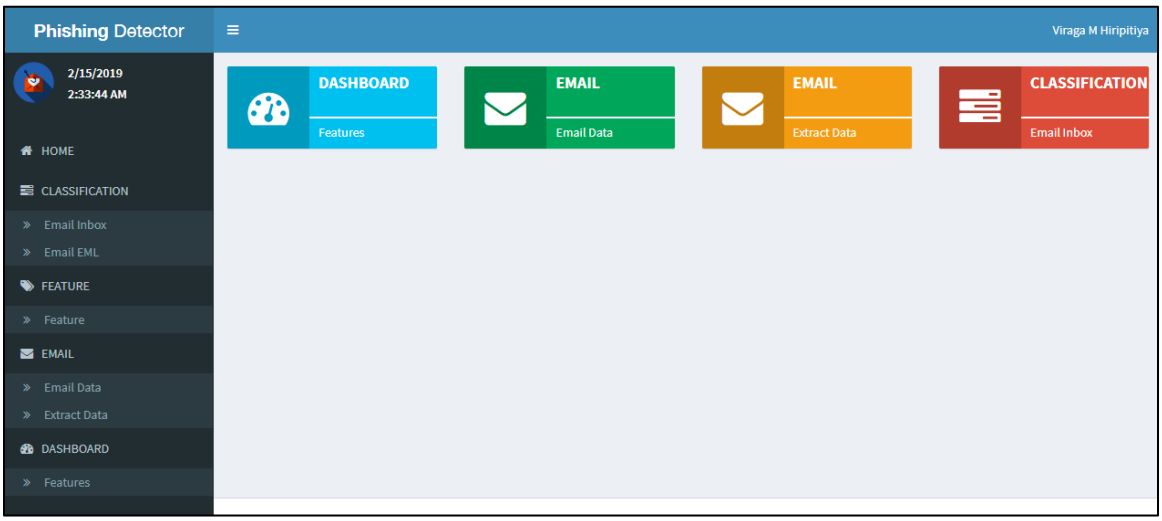


Figure E.1 Home Page

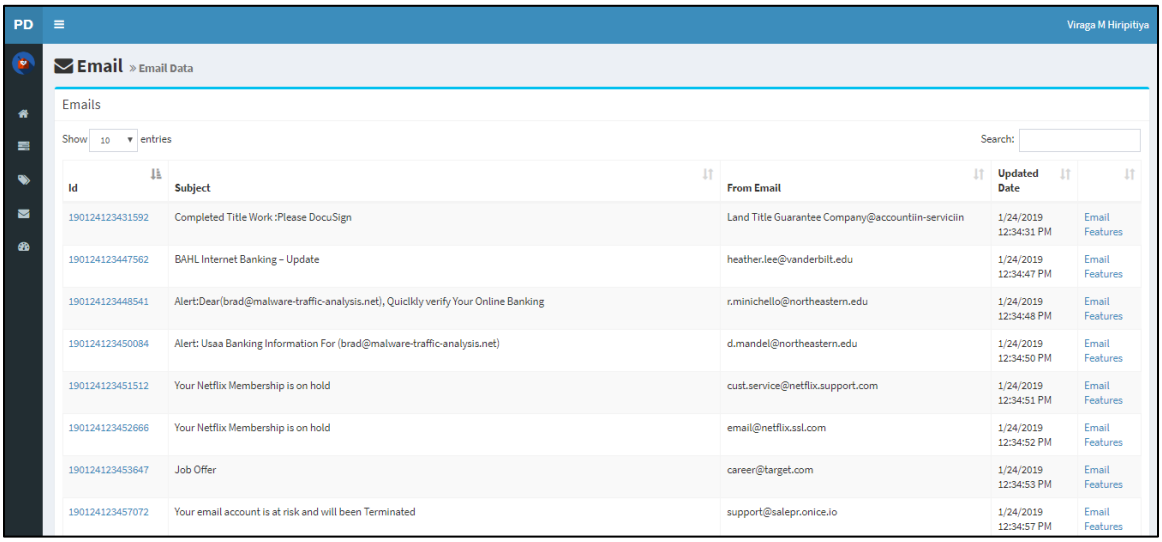


Figure E.2 Email List

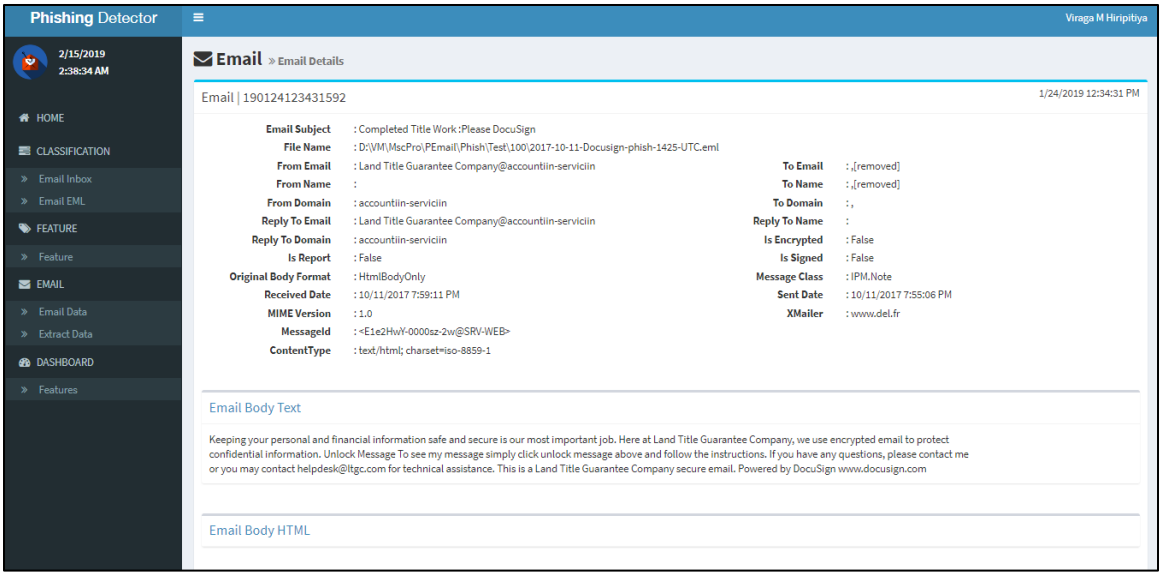


Figure E.3 Email Details

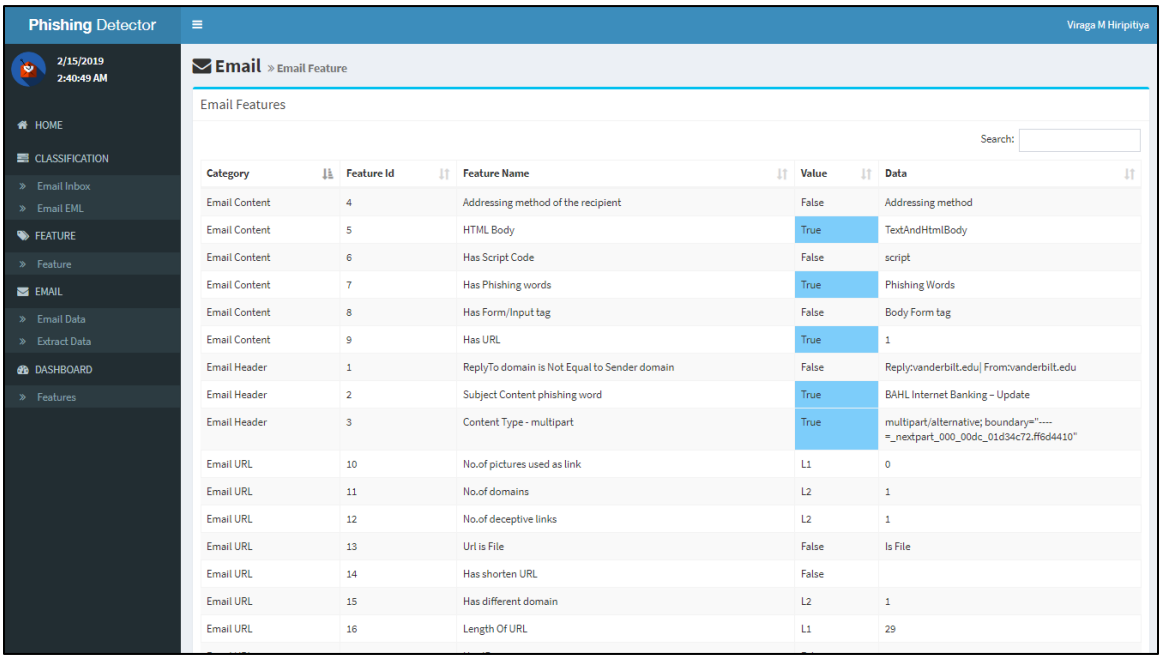


Figure E.4 Email Feature

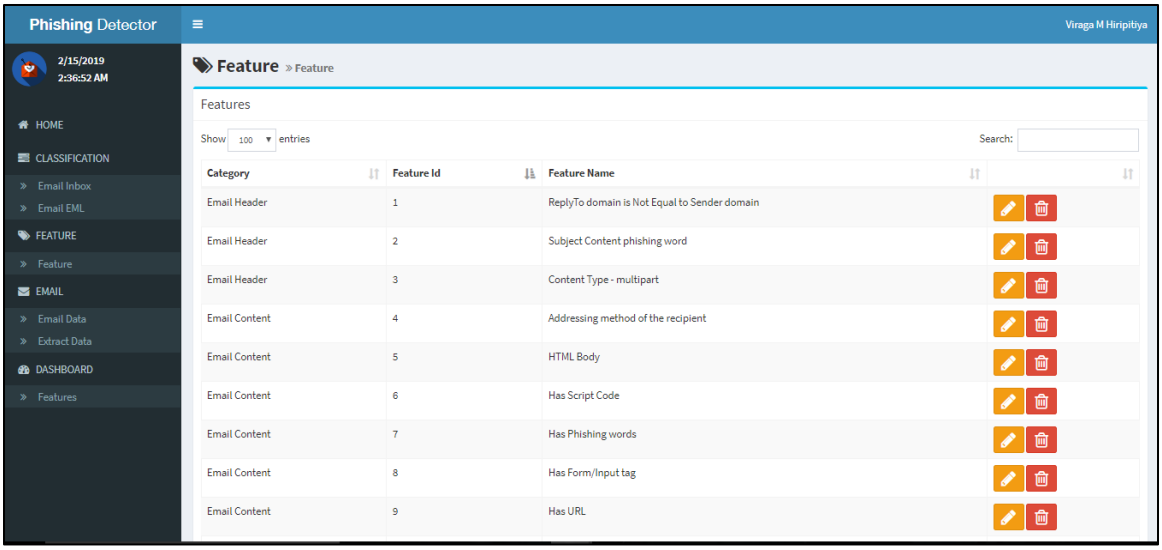


Figure E.5 Features

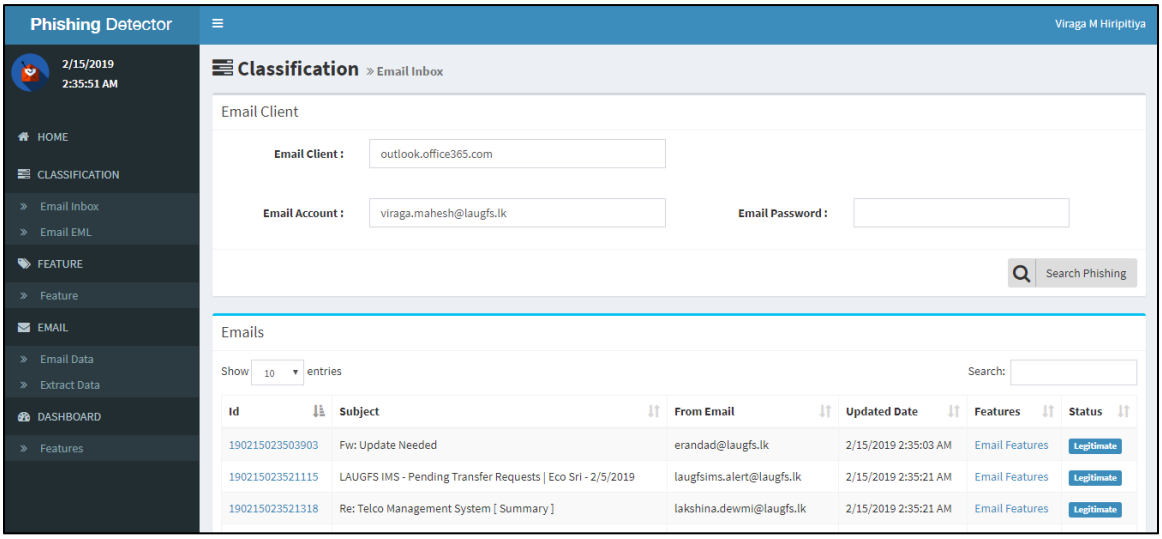


Figure E.6 Classification Inbox Email