

Bachelor of Science in Computer Science & Engineering



**Hybrid Feature Based Phishing Website Detection
using Classification Technique**

by

Sumitra Das Gupta

ID: 1504016

Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)
Chattogram-4349, Bangladesh.

May, 2021

Hybrid Feature Based Phishing Website Detection using Classification Technique



Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

Sumitra Das Gupta

ID: 1504016

Supervised by

Dr. Iqbal H. Sarker

Assistant Professor

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

The thesis titled ‘**Hybrid Feature Based Phishing Website Detection using Classification Technique**’ submitted by ID: 1504016, Session 2019-2020 has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

Chairman

Dr. Iqbal H. Sarker

Assistant Professor

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Member (Ex-Officio)

Dr. Md. Mokammel Haque

Professor & Head

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Member (External)

Muhammad Kamal Hossen

Associate Professor

Department of Computer Science & Engineering

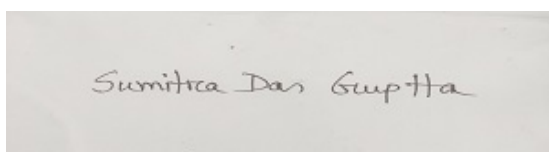
Chittagong University of Engineering & Technology (CUET)

Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.

A rectangular box containing a handwritten signature in cursive script that reads "Sumittra Das Gupta".

Signature of the candidate

Date: 24-5-2021

Acknowledgements

The satisfaction that accompanies the successful completion of this work would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. I am grateful to my honorable project Supervisor Dr. Iqbal H. Sarker, Assistant Professor, Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, for the guidance, inspiration and constructive suggestions which were helpful in the preparation of this project. I would also like to convey my gratitude towards our head of the department Dr. Md. Mokammel Haque. I also convey special thanks and gratitude to all my respected teachers of the department. I would also like to thank my friends and the staffs of the department for their valuable suggestion and assistance that has helped me in successful completion of the project. Finally, I would like to thank my family for their steady love and support during my study period.

Abstract

Phishing is a type of cyber-security attack that uses deception to fool Internet users by disclosing sensitive information such as usernames, passwords, social security numbers, and credit card numbers. Usually attackers try to deceive internet users by masking a webpage as a official genuine webpage to steal personal and financial information. Many anti-phishing solutions have been presented including blacklist or whitelist, heuristic and visual similarity-based methods. Unfortunately, Internet users continue to fall prey to phishing websites and expose important information. Techniques based on blacklists, whitelists, or visual similarities cannot detect zero-hour phishing attacks or brand-new websites. Furthermore, earlier machine learning-based approaches extract features from third-party sources, such as a search engine. As a result, they are complex, sluggish, and unsuitable for real-time environments. To solve this gap, this study proposes a hybrid feature based anti-phishing strategy based on machine learning that just retrieves features from the client side. We collected 15 features from the URL alone and 10 features from the hyperlink information acquired from the page source without relying on a third party to generate hybrid feature set, making the proposed method faster and more trustworthy. A fresh dataset is created for measuring the system's performance, and the experimental results are tested on it. In comparison to existing methods, our proposed methodology has achieved a high test accuracy of 99.17% in detecting phishing websites, with a true positive rate of 98.81% and also a very low false positive ratio of 0.49%.

Keywords— phishing detection, machine learning, hyperlink, URL, classifier, anti-phishing, cyber security, XG Boost.

Table of Contents

Acknowledgements	iii
Abstract	iv
List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Introduction	1
1.2 Design Overview	2
1.3 Difficulties	3
1.4 Applications	3
1.5 Motivation	3
1.6 Contribution of the thesis	4
1.7 Thesis Organization	4
1.8 Conclusion	5
2 Literature Review	6
2.1 Introduction	6
2.2 User Awareness/Education	6
2.3 Software Based	6
2.3.1 List Based Techniques	7
2.3.2 Visual Similarity Based Techniques	7
2.3.3 Machine Learning Based Approach	8
2.4 Related Literature Review	8
2.5 Conclusion	10
3 Methodology	11
3.1 Introduction	11
3.2 Dataset Development	11
3.3 Proposed Approach	13
3.3.1 Feature Extraction	13
3.3.1.1 URL Based Feature	15
3.3.1.2 Hyperlink Based Feature	21

3.3.1.3	Hybrid Features	26
3.3.2	Train Classifier Model	26
3.4	Conclusion	27
4	Results and Discussions	28
4.1	Introduction	28
4.2	Implementation Tools	28
4.3	Experiments	29
4.3.1	Experimental Data	29
4.3.2	Evaluation Measures	29
4.4	Evaluation of Performance	30
4.4.1	Results on Popular Machine Learning Classifier	30
4.4.2	Evaluation of Features	31
4.4.3	Comparison with other Approach	33
4.5	Conclusion	34
5	Conclusion	35
5.1	Conclusion	35
5.2	Future Work	35

List of Figures

1.1	Phishing attack overview in 2020	2
3.1	A Phishtank’s search page with valid phishing URLs searching. . .	12
3.2	A Phishtank’s phish detail page for a phish URL	12
3.3	Proposed Methodology.	14
3.4	A phishing website along with it’s suspicious URL & source code .	14
3.5	Structure of URL(Uniform Resource Locator)	16
3.6	HTML DOM Tree.	21
4.1	Performance comparison of our approach using various classifiers.	30
4.2	ROC curve of XG Boost classifier.	31
4.3	Performance measures based on URL Features.	32
4.4	Performance measures based on Hyperlink features.	32
4.5	Overall performance measures using hybrid features.	32

List of Tables

3.1	Categorical features used in our approach	15
3.2	Optimized parameters for ML models	27
4.1	Comparison between various anti-phishing approaches based on performance	33

Chapter 1

Introduction

1.1 Introduction

Phishing is one of the most serious and dangerous online security threats in the today's world. The number of people using social networks, e-commerce, electronic banking and other online services has been increasing day by day due to the internet availability at lower price and rapid development of global networking and communication technologies. This situation looks like an opportunity to make money for an attacker by stealing confidential information from the internet users [1]. The attacker develops a website that appears to be identical to the real one. This fraudulent website's link is then send to millions of internet users via various social networking and communication sites such as Facebook, Twitter, Email etc. Typically, the fake email content conveys a sense of panic, urgency, or a financial bid, and instructs the recipient to take immediate action [2]. When a user unwittingly clicks on this phishing link and updates any sensitive credentials, cyber attackers gain access to the user's information like financial data, personal information, username, password, etc. This stolen information is used by cyber criminals for a variety of illegal activities, including blackmailing victims. According to [3], there are five reasons why users fall for phishing:

1. Users do not have a deep understanding of URLs.
2. Users are unsure of which websites they should rely on.
3. Due to redirection or secret URLs, users do not able to see the entire address of the web page.
4. Users don't have much time to look up a URL or unconsciously visit certain web pages.

5. Users are unable to differentiate between legal and phishing web sites.

Phishing attacks are now the most common way for other malicious software, such as ransomware, to spread[2]. Anti-Phishing Working Group (APWG), a non-profit organization, investigates the phishing attacks and publishes reports on a regular basis (quarterly and half-yearly). A recent Phishing Activity Trends Report published by APWG (2020) shows that the number of phishing attacks observed by APWG members increased by more than doubling in 2020 (APWG Q4 2020 Report 2020)[4]. During the Covid-19 pandemic, 225,304 new phishing sites were discovered in the month of October alone, smashing all previous monthly records. The growth of phishing attacks is depicted in Figure 1.1 for the year 2020.

Phishing Attacks Doubled in 2020 as October Shatters Monthly Records

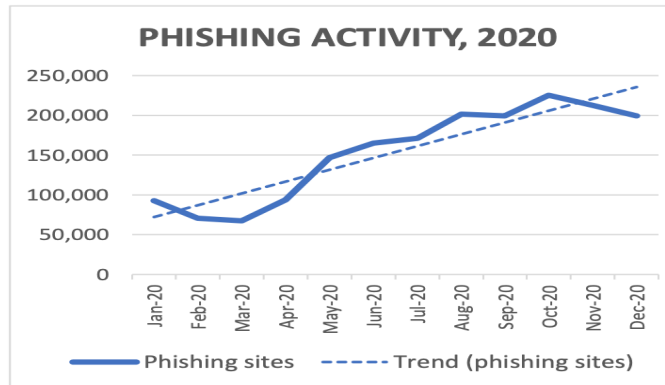


Figure 1.1: Phishing attack overview in 2020

1.2 Design Overview

We propose a hybrid feature based phish detection approach using classification techniques for solving the above problems. Our feature extraction process is not dependent on any search engine or third party services. This thesis work presents an approach to detect phishing websites using hybrid features that include URL based features, also called address bar features and hyperlink based features. Hyperlink informations are extracted from the source code of the webpage and then analyze it. We have collected 15 different features from the URL and 10 different categories of features from the hyperlink informations.

1.3 Difficulties

In order to develop this phishing detection system, we must overcome a number of challenges. Some of these challenges are outlined below.

- We needed to create our own dataset with 6000 URLs because to the lack of a globally accepted test set for phishing systems. There are 3000 valid URLs and 3000 phishing URLs in this collection.
- Phishing websites are short lived. So, we had to crawl them and extract features when they are alive.

1.4 Applications

With the advent of e-commerce, e-banking, social networking and social media, phishing attacks are also on the rise. As a result, the industry and internet users are facing huge financial losses. Security is a big concern for the huge stream of information used in everyday. Therefore, effective steps need to be taken to accurately identify and address phishing attacks in a short period. Our proposed approach is faster and able to detect zero day phishing attacks. It can be used to protect information from phish attackers.

1.5 Motivation

For detecting phishing websites, a number of different methods are used. Machine learning is the most common anti-phishing technique for detecting phishing websites. Recent advances in phishing detection have sparked the development of a number of new machine learning-based techniques. In the machine learning based techniques, a predefined set of features are used for train a classification algorithm that can determine whether or not a website is phishing. Most of the existing phishing detection approaches that are based on machine learning extract those features from the search engine, third party, web traffic, DNS etc. This type of feature extraction is complicated and time consuming. So, they are not fit for the real time phish detection. According to the statistics in the APWG

study 1H2014 [5], phishing sites have a median life cycle of less than 10 hours. This report also specifies that half of the phishing websites were taken down in less than a day. So, for the real-time setting, a quick and intelligent phishing detection solution is needed.

1.6 Contribution of the thesis

Our paper’s key contributions are as follows:

- We collect phishing and legitimate website’s url from the open source platforms and create our own dataset
- Our proposed approach develops a hybrid feature set from hyperlink based and URL based features that doesn’t depend on third party services such as DNS, search engine, Certification Authority etc. Therefore, it can provide better privacy and require shorter time to detect.
- We have proposed a phishing website detection system based on a hybrid (combined) feature set that can detect “zero-hour” phishing attacks with higher accuracy rate.
- Phishing websites written in any textual language can be detected using the proposed method.

1.7 Thesis Organization

The following is how the rest of the report is structured:

- Chapter 2 discusses a number of existing anti-phishing approaches based on machine learning.
- In Chapter 3, we give insight into the details of our proposed strategy. The extractions of various features to train the machine learning algorithms are also presented in this chapter. A detailed discussion about our data crawling and dataset creation is also discussed in depth.

- In Chapter 4, the implementation details, evaluation metrics, and experimental results are discussed with appropriate visualizations.
- Finally, Chapter 5 brings the report to a conclusion and discusses future research.

1.8 Conclusion

In this chapter, we have tried to introduce with a most common and dangerous cyber crime, called phishing attack and give an overview of machine learning based phishing website detection. We also discussed a few challenges faced while implementing the work and motivation behind this work. In the next chapter, background and current state of the problem will be discussed in depth.

Chapter 2

Literature Review

2.1 Introduction

Nowadays, many anti-phishing solutions are proposed by different authors for detecting phishing websites. In this chapter, we try to give a summary of these solutions. Generally, we divide the phishing website detection approaches into 2 categories: User awareness/education based and software based. The following is a summary of these approaches.

2.2 User Awareness/Education

User education approach is mainly depends on how efficiently educate the internet users so that they can understand the characteristics of phishing attacks[6]. This understanding helps them to correctly identify the phishing websites and emails. An interactive teaching game is developed by Sheng et al.[7] called “Anti-Phishing Phill”, which teaches players how to recognize phishing websites. Users who played the game were better able to recognize phishing websites than those who did not. The main goal of this game is to give computer users with conceptual knowledge about phishing assaults.

2.3 Software Based

Software based detection techniques are classified into listing based (blacklist/whitelist), visual-similarity-based, machine learning based approaches.

2.3.1 List Based Techniques

List based detection approaches are two types: blacklist and whitelist techniques. Most of the browsers have their own list of blocked and safe Uniform Resource Locators (URLs). The database containing the blocked URLs is called blacklist and database of unblocked or safe URLs is called whitelist. Wang et al.[8] and Han et al.[9] both utilize a white list-based approach to classify URLs. Chiew et al.[10] worked with logo extraction and matching this logo using whitelist. Other solutions for detecting suspicious URLs employ whitelists of resources such as layouts (Rosiello et al. [11]) and favicons (Chiew et al.[12]).

In the blacklist method, a suspicious domain is checked whether it matches in the blacklisted domains or not. If it is matched, then it is classified as phishing, otherwise legitimate. To detect new phishing URLs, Felegyhazi et al.[13] employed domain name and name server information from blacklisted URLs. The information from a URL's registration and DNS zone is compared to the information that are already stored in the blacklist.

The whitelist approach is exactly the opposite of blacklist method. The main drawback of this list based techniques is that it wrongly classifies the newly generated phishing websites whose age is less than a day, known as zero day phishing attacks.

2.3.2 Visual Similarity Based Techniques

In the visual similarity based techniques, the visual outlook of the suspicious website and its corresponding legitimate website are compared. To compute the similarity across websites, these techniques [14] use a variety of features such as page source code, photos, textual content used in website, HTML codes, CSS(Cascading Style Sheets), website logo, and so on. Most of the time, attackers copy the targeted website's outlook so that users can easily get tricked.

So, in this technique, similarity score is calculated for suspicious site and the

suspicious site is determined as phishing if its similarity score is higher than a certain threshold. This technique is also not so effective for detecting zero hour attacks.

2.3.3 Machine Learning Based Approach

In the machine learning based approach, a dataset is created with extracted features. Then a classification algorithm is trained with feature set and a website is classified as phishing if it matches with the predefined feature set. Machine learning approaches can detect even zero day attacks if it is trained with heuristic features. Over all of the phishing website detection approaches, machine learning approach is the better one. However, to gather and compute features from diverse sources such as address bar(URL), source code, website traffic, DNS, WHOIS database, search engine etc., certain machine learning solutions necessitate large computations.

2.4 Related Literature Review

Our proposed approach is also based on machine learning. Here, some of the machine learning based approaches are discussed below.

Rao et al.[1] used both machine learning approaches and image checking for evaluating their proposed model. They proposed a phishing website detection model based on heuristic features. They extracted their heuristic features from URL, source code and third party services. They also showed how the third party service dependent features improve the accuracy of their model. Their proposed model also able to detect zero day phishing attacks. It also has a performance problem because of using the third party service dependent features.

In [2], a novel anti-phishing approach based on machine learning algorithms was proposed. They used only hyperlink based features that are extracted from the source code of the webpage. Because of using only hyperlink knowledge as features, they claimed that their model is totally third party independent and language independent. It also has a problem of wrongly classifying the non html

websites.

O. Sahingoz et al.[3] identified phishing websites by classifying them with NLP based features and word vectors. NLP based features are those which are mainly determined by the human eyes. They constructed a new dataset with such a big amount of data and achieved 97.98% high accuracy rate using Random Forest algorithm for detecting phishing URLs. However, their proposed system can not detect the phishing websites which have shorter url.

Y. Huang et al. [15] mainly used capsule based neural network which consists of four branches.They used one convolution layer for shallow features extraction from the URLs and then used two capsule layers for accurate representation of URLs from those shallow features.Their proposed model performed very well with high accuray rate ,but the design architecture of the network is quiet complex.

Rao et al. [16] developed a light-weight feature based application named Catch-Phish that differentiate between phishing and legitimate website without visiting the website. They used two categories of features: hand-crafted features that are actually URL based and TF-IDF features .This application achieved 94.26% accuracy using random forest classifier on their collected dataset. It can be used as a first level filtering of phishing websites within shorter time period.

Odeh et al. [17] achieved a very high accuracy rate approximately 99% using applying adaptive boosting approach. They have collected 30 features and these features are classified into four categories: Address bar based, abnormal based, HTML and JavaScript based and domain based features. They took the mostly correlated features by utilizing features' selection. Though they achieved a high accuracy rate, their approach is not fit for the real time phish detection.

Babagoli et al. [18] used a non-linear regression strategy for detecting a website whether it is phishing or not. They used a large dataset with 11055 webpages. Two types of meta-heuristic algorithms, Harmonic search and SVM are used for model training. This study concluded that HS gives better performance than SVM and they achieved 92.80% accuracy using HS algorithm.

R. Mohammad et al. [19] implemented a phishing attack detection model using the self-structuring neural network. Authors used the back propagation algorithm for weight adjustment of the network. They used 17 features collected from the

URL, source code of website and also third party services. The detection time is increased for using the third party based features. However, their test set accuracy was 92.18% with 1000 epochs.

F. Feng et al. [20] designed a novel neural network for phishing detection. They designed their proposed novel neural network in such a way that the design risk is minimized and used the Monte Carlo (MC) algorithm for training process. However, 97.71% accuracy was achieved by them with a low false positive rate of 1.7%. They also showed that their proposed model performs better than other machine learning classifier.

Usually, the performance of the machine learning algorithms depends on the quality of feature set and the training data size. The accuracy also depends on which features are extracted and selected for training the model.

2.5 Conclusion

This chapter contains a thorough summary of the literature review. We briefly discussed on previous works that is already implemented, their limitations and their role on website classification applying various anti-phishing techniques. the researchers used a range of feature extraction and classification algorithms. The methodology for the whole system is thoroughly explained in the following chapter.

Chapter 3

Methodology

3.1 Introduction

This chapter presents the proposed methodology for phishing website detection and explains in details with its constituents. The dataset collection and development procedure also presented in this Chapter. We also go through the feature extraction and each feature is described in depth. The detailed description of the training model preparation, feature extraction included in Chapter 3.

3.2 Dataset Development

We have extracted our desired features from 6000 different legitimate and phishing websites. Phishing websites have a limited lifespan. So, we crawled when they are alive. Phishing URLs are collected from Phishtank[21]. We have developed a phish crawler to crawl the phishing URLs from the phishtank website. There is a 'Phish Search' tab on the Phishtank website. In this section, we can quickly search for all of the current valid phishing URLs. The 'ID' of the URL in the Phishtank database, the URL itself, the 'validity' property (search for 'valid phish'), and the 'online' properties are all shown on this page in a table. Each search page contains 20 URLs. The URL address of this search page includes the search type ('valid phish') and page number. A search page is shown in figure 3.1. By looping the page number, a request is sent for the search page and BeautifulSoup module is used for accessing the source code of this page. From this, phishing IDs are collected from each page's table, not the URLs. As the end of long URLs in the table is '...', we don't get the full URL if we extract it from the table content. Following that, these 'IDs' are searched in Phishtank's 'phish detail' section that

is shown in Figure 3.2. A request is then sent for accessing the page source of this phish detail pages and from this, we extract our desired valid phishing URL. Using this phish crawler, we have crawled 3000 phishing URLs. We have collected the legitimate /benign URLs from the dataset provided by University of New Brunswick[22]. This dataset contains over 35,300 legitimate URLs that were collected from the Alexa top websites[23]. From this dataset, 3000 legitimate URLs are picked randomly for our experiment. Labelled values are set to 0 for legitimate URLs and 1 for phishing URLs.

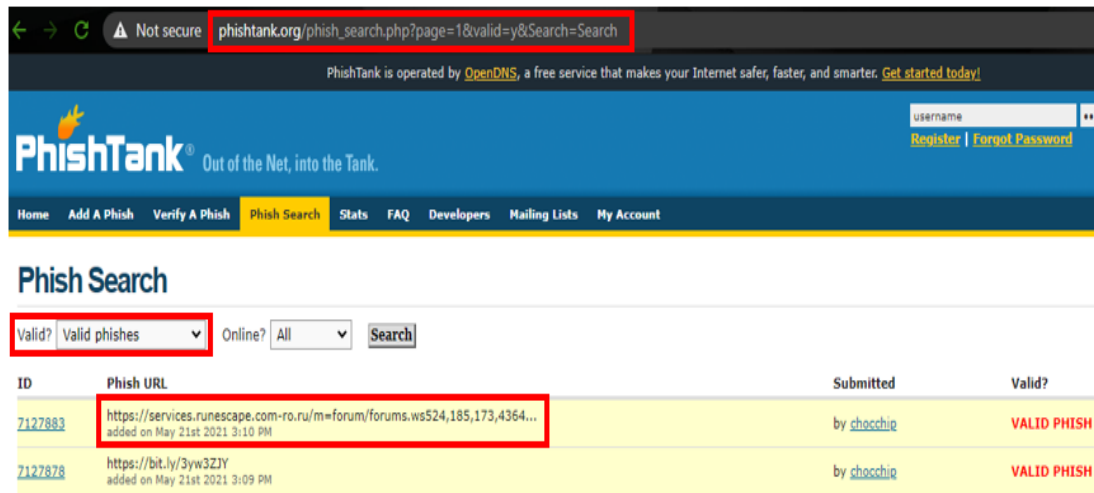


Figure 3.1: A Phishtank's search page with valid phishing URLs searching.

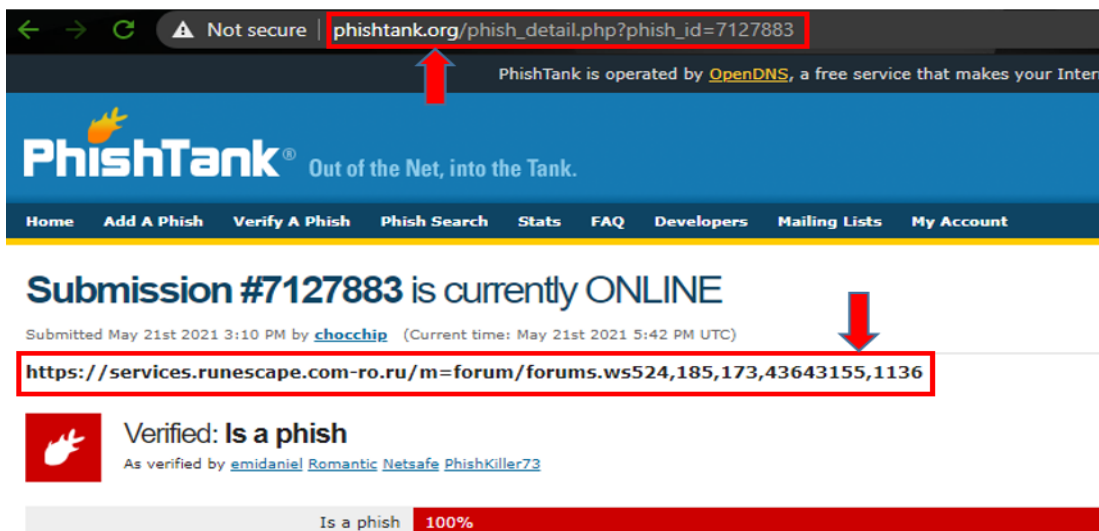


Figure 3.2: A Phishtank's phish detail page for a phish URL

3.3 Proposed Approach

The system architecture of our proposed approach is showed in Figure 3.3. For effective detection of a wide-ranging phishing attack, our proposed approach extracts and analyzes various features of suspicious webpages. We select two categories of features to extract. Our features are based on URL and hyperlink of the webpage. URL based features are extracted by analyzing the structure of the URL. In order to retrieve hyperlink features, source code of the website is first extracted and from this, a Document Object Model (DOM) is generated. This DOM tree is used for extracting the hyperlink information. DOM is an organized tree-like representation of any XML or HTML document. It is easier to search for any hyperlink related information from DOM tree rather than directly searching in the source code of the website. The features are taken from different approaches proposed by different authors, but we modified some of them for better results. All the features are explained in detail in the upcoming subsection. The obtained features are combined together to create a hybrid feature set and this hybrid feature set is used to train the model for website classification using various machine learning algorithms. By evaluating the performance of different classification algorithms, we find out the best classifier that can classify a input URL either phishing or legitimate with a lower error rate and higher accuracy.

3.3.1 Feature Extraction

Labelled raw URLs converted to embedding feature representation as the training cannot be performed on strings. So, we have developed a feature extractor that extracts various features from the raw URL text. Our used features are divided into two categories as shown in Table 3.1. We have used 25 features (categorized into two groups: URL features, hyperlink features) for classification of webpages. Because of the limitation of search engine and third party dependent approaches, we use only the client side specific features. Some of the feature value is in the form of 0 or 1, where 0 indicates legitimate and 1 indicates phishing. In figure 3.4, a phish site that exactly looks like the login page of the official paypal website have shown. We highlight some of the unique phishing indication features also.

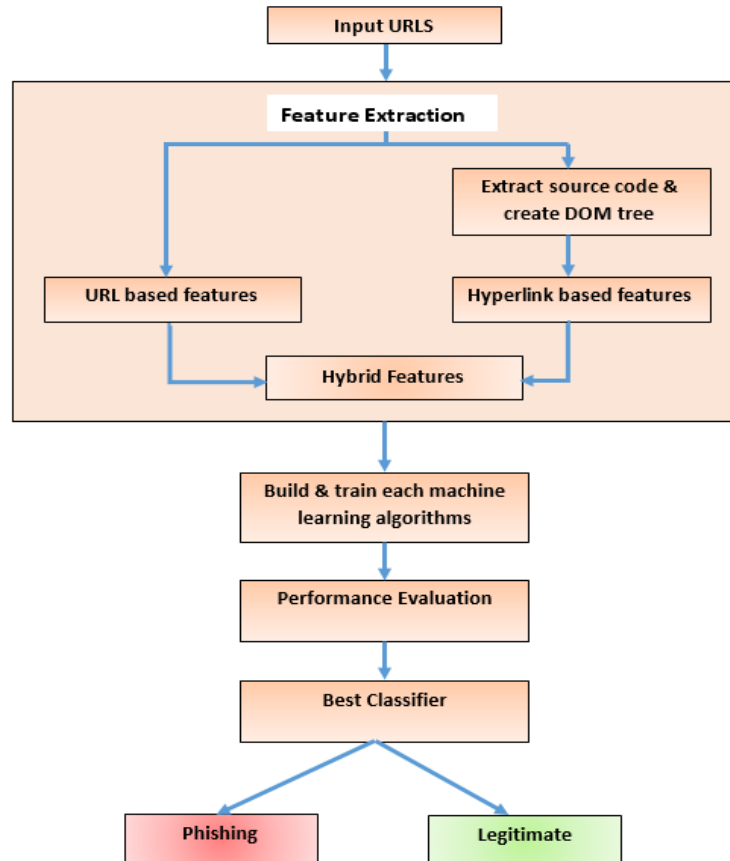


Figure 3.3: Proposed Methodology.

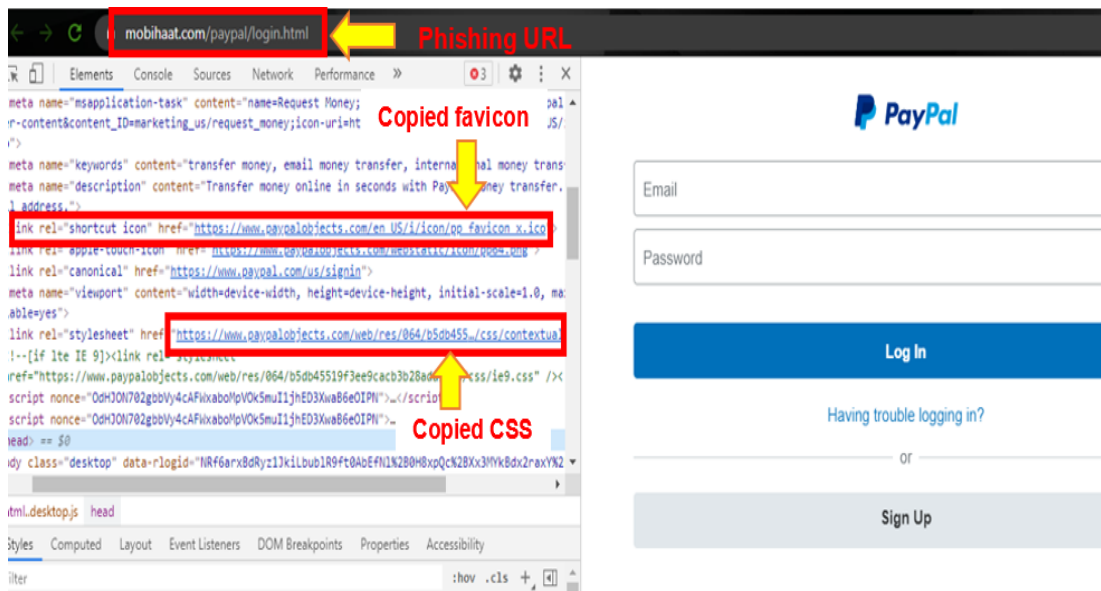


Figure 3.4: A phishing website along with it's suspicious URL & source code

Table 3.1: Categorical features used in our approach

Serial No	Cat-egory	Features Name	Total Fea-tures
1	URL based features	Domain of URL, Count Subdomains in URL, IP Address in URL, "@" Symbol in URL, Length of URL, Depth of URL, Redirection "/" in URL, "http/https" in Domain name, HTTPS_scheme, Using URL Shortening Services "TinyURL", Prefix or Suffix "-" in Domain, Existence of sensitive word, Existence of trendy brand name, Existence of upper case letter, Number of dots in url	15
2	Hyper-link based features	No hyperlink, Internal Hyperlink ratio, External hyperlink ratio, Internal/external CSS, Suspicious Form link action, Null hyperlink, Internal/External Favicon, Common page detection ratio, Common page in footer section ratio, Server Form Handler	10

The name of the features and their description along with the criteria for assigning values to the features are described in the following.

3.3.1.1 URL Based Feature

Uniform Resource Locator (URL) is used to find things in the internet such as photographs, audio or video files, hypertext pages etc. Figure 3.5 represents the structure of URL. The structure of a URL is divided into various components. The URL starts with a protocol like http, https, ftp etc. Protocol is used to access the resource on web. The more safe protocol is HTTPS (Hypertext Transfer Protocol Secure). The second component indicates the location where the resource are placed. It is hostname or sometimes an IP address. Hostname is divided into three sub-components: subdomain, primary domain and top-level domain (TLD). TLD is further subdivided into several components such as generic TLDs (gTLD), country-code TLD (ccTLD) etc. The third component of the URL is path that indicates the specific resource requested by the user to access within the domain. The domain part and the path is separated by a single slash '/'. The path structure has two optional fields. The first one is query that is always starts

with a question mark ‘?’ and another one is fragment that is preceded by hash ‘#’. The standardized format of URL is as follows:

<Protocol>://<Sub domain>.<Primary domain>.<TLD>/<Path domain>
<?query><#fragment>

A phisher has complete control over the subdomain, primary domain, and path segment values. In this subsection, we explain how a cybercriminal uses the URL obfuscation trick to deceive users.

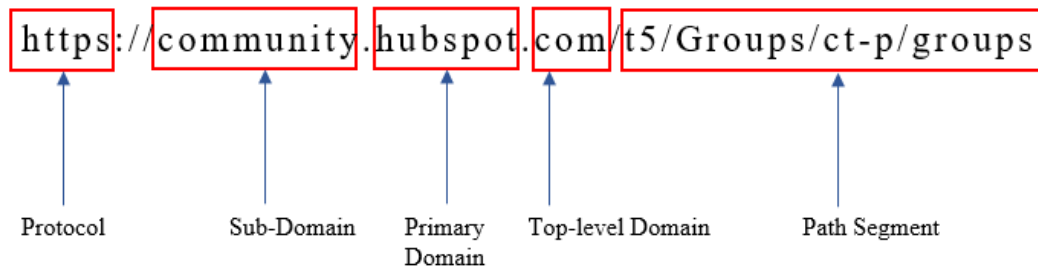


Figure 3.5: Structure of URL(Uniform Resource Locator)

Domain name of URL

We have stored the entire domain name except “www.” from the URL as a feature. This feature isn’t particularly useful for implementation. It’ll be removed from the feature set while training the model.

UF1 = domain name of URL

Count subdomain in URL

This feature checks the number of dots in the hostname part of a URL. Generally a legitimate URL has two dots in the domain part except ‘www.’. Phishers add more subdomains and also add the domain of real websites as a subdomain in the phishing website to deceive the users. They add dots to add more subdomains in phishing URL. If the number of dots in hostname is equal to three, the URL is ‘suspicious’ and feature value is set to 0.5. If the number of dots is greater than

three, URL has multiple subdomains, so it is classified as ‘phishy’ and feature value is set to 1.

$$UF2 = \begin{cases} 1, & \text{if dots in URL} > 3 \\ 0.5, & \text{if dots in URL} == 3 \\ 0, & \text{otherwise} \end{cases}$$

IP Address in Domain

Sometimes, attackers use IP address as an alternative of domain name in the domain part of URL. If IP address of any version (IPV4 or IPV6) is found in the domain part of the URL, it ensures that someone is definitely attempting to steal user’s personal informations.

$$UF3 = \begin{cases} 1, & \text{if IP present} \\ 0, & \text{otherwise} \end{cases}$$

“@” symbol in URL

The overall URL is analyzed to see if it has the special symbol "@". Generally attackers add “@” after the actual domain of the website so that it seems like a legitimate website’s domain to the users. When a user clicks on this link, the browser ignores everything before the “@” symbol and downloads the real address that comes after it. The feature value is set to 1 for the presence of “@” in URL else set to 0.

$$UF4 = \begin{cases} 1, & \text{if '@' present} \\ 0, & \text{otherwise} \end{cases}$$

Length of URL

Attackers make their phishing URLs long in order to hide any suspicious parts of the URL. Larger the number of characters in URL, higher the possibility of website to be a phishing one. Scientifically, there is no exact range for length that separates phishing URLs from legitimate URLs. In [2], their proposed length of a legitimate URL is 75. If the length is more than 75 but less than 100, then we consider the URL as suspicious and assign the feature value 0.5.

$$UF5 = \begin{cases} 0, & \text{if URL length} < 75 \\ 0.5, & \text{if URL length} \geq 75 \text{ and URL length} < 100 \\ 1, & \text{otherwise} \end{cases}$$

Depth of URL

This feature counts the number of subpages in the URL of a website. The subpages are separated by a single slash (“/”) symbol in the path segment of the URL. The file directory of the legitimate website is not so much longer. Attackers, on the other hand, save their phish pages in their phish computers at a very deep level. So, their webpage’s file path is usually longer than authentic one. The feature value is numerical based on the URL.

UF6 = total number of subpages/ subfolders in the URL path

Redirection “//” in URL

The symbol “//”, known as double slash is used for redirecting the user to other website. When generating the URL address, phishers use this slash symbol to redirect the user to a fake website. For this feature, “//” location in the URL is calculated. This location returns the last appeared location of “//” in the entire URL. This symbol appears in the 6th or 7th position (after http: or https:) in genuine cases. So, if the location is greater than 7, then we can say that “//” appears somewhere in the URL other than after the protocol. This feature is a binary feature with value 1(phishing) or 0(legitimate).

$$UF7 = \begin{cases} 1, & \text{if " //" is found anywhere other than after the http/https protocol} \\ 0, & \text{otherwise} \end{cases}$$

“http/https” in domain name

In order to confuse users, phishers can add the “HTTPS” or “HTTP” token to the domain portion of a URL. In this feature, the existence of "http/https" in the domain portion of the URL is investigated. If the domain portion of the URL contains "http/https," the value assigned to this feature is 1 (phishing), otherwise

0 (legitimate).

$$UF8 = \begin{cases} 1, & \text{if http/https exists in domain portion of URL} \\ 0, & \text{otherwise} \end{cases}$$

https in scheme

This feature checks the protocol of the URL. If the protocol of the URL is “HTTPS”, then the feature value is assigned to 0(legitimate) or otherwise 1(phishing). The protocol is a very important portion in a URL. Most of the legitimate websites use the HTTPS protocol for more secure connection when confidential information must be forwarded. But nowadays, phishers also to use the fake HTTPS connection to trick the users. So, this feature has not so much effect is detecting phishing websites from legitimate one.

$$UF9 = \begin{cases} 1, & \text{if URL protocol is https} \\ 0, & \text{otherwise} \end{cases}$$

Using URL shortening service “tinyURL”

On the World Wide Web, URL shortening is a technique for making a Uniform Resource Locator (URL) significantly shorter while still connecting to the desired page. This is accomplished by the use of a redirect, which points to a web page with a long URL. For example, the URL "https://en.wikipedia.org/wiki/URL shortening" can be shortened to “https://w.wiki/U”. But at this time, many phishers use the URL shortening services to deceive users. SO, in this feature, we check whether the URL is using the URL shortening services or not.

$$UF10 = \begin{cases} 1, & \text{if tiny URL exists} \\ 0, & \text{otherwise} \end{cases}$$

Prefix or Suffix “-” in domain

This feature checks for the presence of any “-” in the domain part of URL. In legitimate domain names, dashes are seldom used. In order to fool the victim, attackers usually resort to add prefixes or suffixes separated by a dash line “-” to the domain names. This address appears to be authentic to users, and they are

trapped as they try to access this phishing website.

$$UF11 = \begin{cases} 1, & \text{if domain part includes "-"} \\ 0, & \text{otherwise} \end{cases}$$

Existence of sensitive word

Some of the words or tokens are commonly used in the phishing urls such as ‘login’, ‘update’, ‘validate’, ‘activate’, ‘secure’ etc. This type of words are used in the URL to convey urgency to users, encouraging them to visit the phishing site right away. We came up with a list of 18 such phish term. If given URL contains any one of the sensitive phishy word from the list, the feature value is assigned to 1(phishing), else set to 0(legitimate).

$$UF12 = \begin{cases} 1, & \text{if any sensitive word in URL} \\ 0, & \text{otherwise} \end{cases}$$

Existence of trendy brand name

Phishing websites are often produced by imitating well-known brand websites. As a result, hackers use trendy brand names in phishing URLs to easily fool users. When a user sees the brand name in the URL, he assumes it is the brand’s official website URL. We’ve compiled a list of 19 top-level brand names that phishers commonly target.

$$UF13 = \begin{cases} 1, & \text{if any trendy brand name exists in URL} \\ 0, & \text{otherwise} \end{cases}$$

Existence of upper case letter

Legitimate URLs are usually written in lowercase letters only. But phishing URLs often use uppercase letters for fooling users. If URL has any upper case letter, then it is assumed to be a phishing URL.

$$UF14 = \begin{cases} 1, & \text{if any uppercase letter in URL} \\ 0, & \text{otherwise} \end{cases}$$

Number of dots in URL

There are normally no more than two dots in a genuine website's URL except the 'www.'. So, in this feature, the number of dots in the whole URL is checked. If it is greater than 2, then this URL is classified as phishing, otherwise legitimate.

$$UF15 = \begin{cases} 1, & \text{if no.of dots} > 2 \\ 0, & \text{otherwise} \end{cases}$$

3.3.1.2 Hyperlink Based Feature

In this segment, we'll go through the features that can be extracted from hyperlinks in a website's source code. Hyperlink information is examined from the DOM tree representation of a webpage. The Document Object Model (DOM) is a method of representing a webpage in an organized hierarchical manner so that users may navigate the document more easily. DOM allows us to easily access and manipulate tags, IDs, classes, attributes, and elements. For this feature extraction, we mostly look at link, form, src, and anchor tags. In Figure 3.6, a DOM tree is shown. We have extracted 10 hyperlink based features.

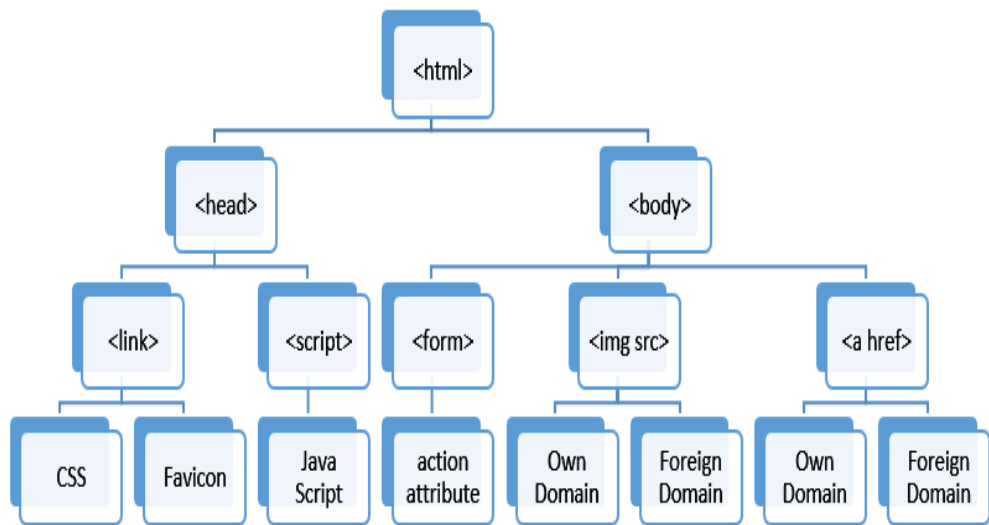


Figure 3.6: HTML DOM Tree.

No Hyperlink

An authentic website would typically have a large number of webpages. On the other hand, phishing websites are small and have a few number of webpages. In addition, if the attackers use the hyperlink hidden strategy, the phishing website doesn't provide any hyperlinks[24]. If a website is authentic, we can find at least one hyperlink in the source code of this website [2]. The total number of hyperlinks are determined by considering href, link and src tags. If total hyperlink count is 0, then this feature value is set to 1(phishing), else set to 0(legitimate).

$$HF1 = \begin{cases} 0, & \text{total hyperlink} > 0 \\ 1, & \text{total hyperlink} = 0 \end{cases}$$

Internal Hyperlink ratio

Internal link of a website means the link pointing to the local /base domain. Phishers usually create the phishing websites that visually looks like the original websites. So, they copy the source code of the official targeted website to easily generate phishing site. As they copy the source code from legitimate one, it may contain many hyperlinks pointing to the targeted website. The majority of hyperlinks on a legitimate website have the same base domain, while many hyperlinks on a phishing site which have the domain of the corresponding legitimate website. For this feature calculation, we compute the ratio of internal hyperlinks with respect to total links found in the source code. If the ratio is greater than or equal to 0.5, the website is assumed to be a valid one and set the feature value to 0, otherwise set to 1.

$$Ratioofinternallink = \begin{cases} \frac{totalinternalhyperlink}{totalhyperlink}, & \text{if total hyperlink} > 0 \\ 0, & \text{total hyperlink} = 0 \end{cases}$$

$$HF2 = \begin{cases} 1, & \text{if Ratio of internal link} > 0.5 \\ 0, & \text{otherwise} \end{cases}$$

External Hyperlink ratio

External hyperlinks refer to the links that have different base domain or foreign domain. Most of the links in a phishing website is external. On the contrary, legitimate websites use less external hyperlinks. So, we calculate the ratio of external hyperlinks with respect to total hyperlinks present in the website source code. As legitimate websites contain less external links, the external hyperlink ratio for a legitimate website is usually small. If the ratio is less than 0.5, this feature value is set to 0, else 1.

$$Ratioofexternallink = \begin{cases} \frac{totalexternalhyperlink}{totalhyperlink}, & \text{if total hyperlink} > 0 \\ 0, & \text{total hyperlink} = 0 \end{cases}$$

$$HF3 = \begin{cases} 1, & \text{if Ratio of external link} > 0.5 \\ 0, & \text{otherwise} \end{cases}$$

Internal/External CSS

CSS (Cascading Style Sheets) is a language for representing document formatting and influencing the visual appearance of sites written in HTML, XHTML, and XML. Phishing websites are generated by imitating the design and visual appearance of the original websites to attract the potential victim. Creating a phishing site is usually a low-effort endeavor for attackers. As a result, rather than developing their own CSS file, they attempt to use the CSS file of the legitimate website they are targeting. CSS is two types: internal and external. External CSS file links are added by using <link> tag. So, for getting the external CSS file links, we search for the <link> tag that has at least two attributes as rel = 'stylesheet' and href = 'url of the css file'. Internal CSS are written inside the HTML code of the websites. Attackers mainly used the external CSS files of official sites to develop phishing websites. So, for setting the feature value, we check if there are any external CSS file found in the source code of the website. If the external CSS file is a foreign/external hyperlink, the feature value is set to 1, otherwise 0.

$$HF4 = \begin{cases} 1, & \text{if CSS file is external and current domain=base domain} \\ 0, & \text{otherwise} \end{cases}$$

Suspicious Form Link Action

Phishing websites normally have a login or sign-in form for getting the personal and financial details of the victims. When any user unknowingly fill up this form in a fake website, all the information entered into the form is transferred to the attackers. The action field of <form> tag typically contains the current website's URL for genuine website. But the action field of a fake login form either contains external links or a PHP file [2]. Sometimes the action field is null or contains '#', 'javascript:void()' etc. So, for checking the authenticity of the login form, we check the action field value of the <form> tag. This feature value is binary.

$$HF5 = \begin{cases} 1, & \text{if external link or php file or " ", "\#",} \\ & \text{"javascript:void(0)" in action field} \\ 0, & \text{otherwise} \end{cases}$$

Null Hyperlink

For this feature, we consider only the anchor tag <a>. This feature calculates the percentage of Null links in a website compared to the number of anchor links. An attacker's plan is to keep the internet user on the same page until he enters sensitive information. As a result, when a user clicks on any of the links on the login page, he is taken back to the same login page. The following code is used to do this.

< a href = "#" >

< a href = "#content">

< a href = "javascript:void(0)">

If the ratio of null anchor links to the total anchor links is greater than 0.34, the feature results in 1, otherwise, the result is 0.

$$Ratioofnullanchorlink = \begin{cases} \frac{totalnullanchorlink}{totalanchorlink}, & \text{if total anchor link} > 0 \\ 0, & \text{total anchor link} = 0 \end{cases}$$

$$HF6 = \begin{cases} 1, & \text{if Ratio of null anchor link} > 0.34 \\ 0, & \text{otherwise} \end{cases}$$

Internal/External Favicon

A favicon is a little icon that serves as a website's identity. Favicon is added by using <link> tag. If favicon shown in the address bar belongs to a foreign domain, the website is considered to be a phishing one. Because phishers often copy official website's favicon. Many users are duped by the bogus website's address bar displaying a duplicated favicon. So, we analyze the <link> tag containing favicon and check the link whether it belongs to the same domain or not. If it is internal favicon, then feature value is 0 (legitimate), else set to 1 (phishing).

$$HF7 = \begin{cases} 1, & \text{if external favicon} \\ 0, & \text{otherwise} \end{cases}$$

Common Page Detection ratio

Attackers put in less work and time to set up fraudulent websites quickly. They populate the website with several anchor links to make it appear legitimate. But they don't actually develop many webpages to add them in the anchor links. Phishers may redirect a few or all of the links to a common page. In phishing sites, this type of scenario leads to a high rate of common page detection. So, in this feature, we measure the ratio of frequency of the site's most common anchor link to the total number of anchor links.

$$HF8 = \begin{cases} \frac{freq.ofmostcommonanchorlink}{totalanchorlink}, & \text{if total anchor link} > 0 \\ 0, & \text{total anchor link} = 0 \end{cases}$$

Common Page in Footer section ratio

This feature is the same as the previous one, but only highlights common page

detection in the footer section.

$$HF9 = \begin{cases} \frac{freq.of most common link in footer}{total anchor link in footer}, & \text{if total anchor link} > 0 \\ 0, & \text{total anchor link} = 0 \end{cases}$$

Server Form Handler(SFH)

If SFH contains an empty or 'about:blank' string, it can be assumed that a phishing attempt. Because an action must be taken based on the information submitted by user. Furthermore, if the SFHs domain name is a foreign domain, this indicates that the website is suspicious. So, we set the feature value to 0.5 for suspicious webpage, 0 for legitimate and set 1 for phishing one. This is a ternary feature.

$$HF10 = \begin{cases} 1, & \text{if SFH contains " " or "about:blank"} \\ 0.5, & \text{else if SFHs domain name is foreign domain} \\ 0, & \text{otherwise} \end{cases}$$

3.3.1.3 Hybrid Features

To increase the accuracy, we combined both categories of features (URL based and Hyperlink based) to get a hybrid feature set. In total, we have got 25 features.

Before splitting the dataset into train and test dataset, as a part of preprocessing step, the domain name feature (UF1) is removed from the feature set, because it is not a numerical feature and this feature has no significance for classifying a website into phishing or not.

3.3.2 Train Classifier Model

We divided the dataset into train and test set. 80% dataset is taken as train set and remaining 20% dataset is taken as test set. Labelled train set is used for train the model and test set is used to evaluate the performance of each classifier

and find out the best classifier. Five popular ML-based techniques are used to perform the classification task such as Random Forest, Decision tree, Support Vector Machine, Logistic Regression, XG Boost.

A summary of the parameters chosen for ML models are provided in Table 3.2

Table 3.2: Optimized parameters for ML models

| Classifier | Parameters |
|------------------------|---|
| Logistic Regression | solver='lbfgs', C=1.0, max_iter = 100, penalty = 'l2' |
| Decision Tree | criterion='gini', max_depth=5 |
| Support Vector Machine | kernel='linear', C=10, random_state=12 |
| Random Forest | n_estimators=10, random_state=42 |
| XG Boost | learning_rate=0.4, max_depth=5 |

3.4 Conclusion

This chapter goes into the technique for phishing website detection from website URL. The suggested method has been tested for various machine learning techniques. This experiment yielded a hybrid feature based phishing detection architecture that is more faster and efficient approach than existing anti-phishing approaches. The experimental result analysis of the proposed methodology is discussed in the following chapter.

Chapter 4

Results and Discussions

4.1 Introduction

Chapter 4 explains the details experimental analysis on developed dataset for various machine learning methods. This Chapter also investigates the performance of implemented models with various evaluation measures (such as precision, recall, accuracy, f1-score). The details comparative analysis among various methods with existing techniques also illustrated in this Chapter.

4.2 Implementation Tools

A laptop machine having core i3 processor with 2.0 GHz clock speed and 4 GB RAM is used for implementing this phishing detection approach. We implemented our proposed approach using Python Programming Language due to its huge support of using libraries and shorter compile time. Different functions are created for different feature extraction. The extraction process necessitates the use of many libraries. Some of the libraries that are used in the code are given below.

re: This library is used for performing the regular expression operations such as finding the desired string from the URL.

urllib2: To get the response object from any URL and parse the URL components, we used this library.

BeautifulSoup: This library is so useful. This library is used to extract information from HTML and XML documents and to create DOM (Document Object Model).

Favicon: Website's favicon address is extracted by using this python library.

4.3 Experiments

4.3.1 Experimental Data

The total dataset was divided into 2 portions with 4800 and 1800 data as train, and test, respectively. In the test data portion, there are 613 legitimate URLs and 587 phishing URLs.

4.3.2 Evaluation Measures

To analyze the efficiency of the proposed solution, we use the true positive rate, false positive rate, true negative rate, false negative rate, f1 score, accuracy, precision, and AUC(Area Under the Curve).

True-Positive Rate: It computes the ratio of phishing websites correctly detected out of total phishing websites. It is also known as Recall.

False-Positive Rate: It computes the ratio of legitimate websites wrongly detected as phishing with respect to total legitimate websites.

False-Negative Rate: It measures the rate of phishing websites wrongly classified out of total phishing websites

True-Negative Rate: It measures the rate of legitimate websites correctly classified out of total legitimate websites.

Accuracy: It determines the overall rate of accurate predictions

Precision: It computes the ratio of correctly detected phishing websites with respect to all the websites detected as phishing.

F1 Score: It measures the harmonic mean of Precision and Recall.

False Positive Rate(FPR): It is defined as follows:

$$\text{FPR} = \text{FP}/(\text{FP} + \text{TN})$$

ROC Curve: A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.

Area Under the Curve (AUC): That is, AUC measures the entire two dimensional area underneath the entire ROC curve. AUC provides an aggregate measure of performance across all possible classification thresholds.

4.4 Evaluation of Performance

4.4.1 Results on Popular Machine Learning Classifier

We have used 5 different classification algorithms (Random Forest, Decision tree, Support Vector Machine, Logistic Regression, XG Boost) and then compared their performance in terms of true positive rate, false positive rate, accuracy and false negative rate. Performance comparison is shown in figure 4.1. We also looked at the area under the ROC curve (Receiver Operating Characteristic) for a better precision metric. From the comparison chart in 4.1, we analyzed that XG Boost classifier performs best with highest detection accuracy and TPR and lowest FPR and FNR. In our experiment, the area under the ROC curve for phishing website detection is 99.89% produced by XG Boost classifier. The ROC curve for XG Boost is shown in Figure 4.2.

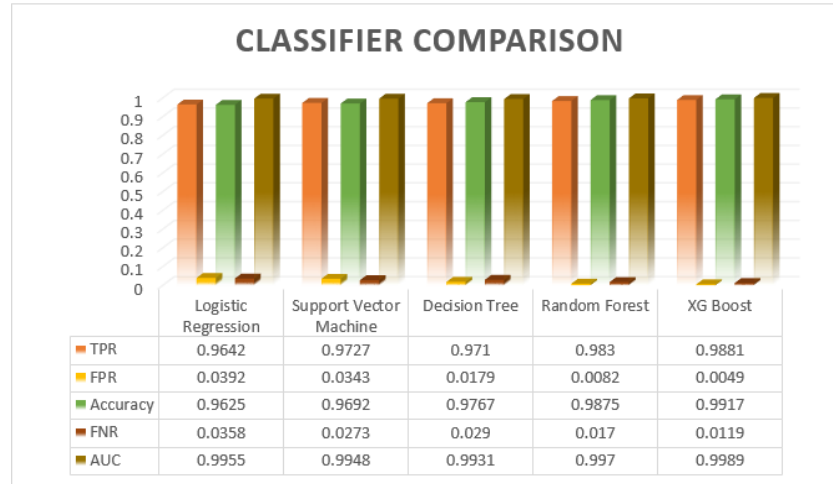


Figure 4.1: Performance comparison of our approach using various classifiers.

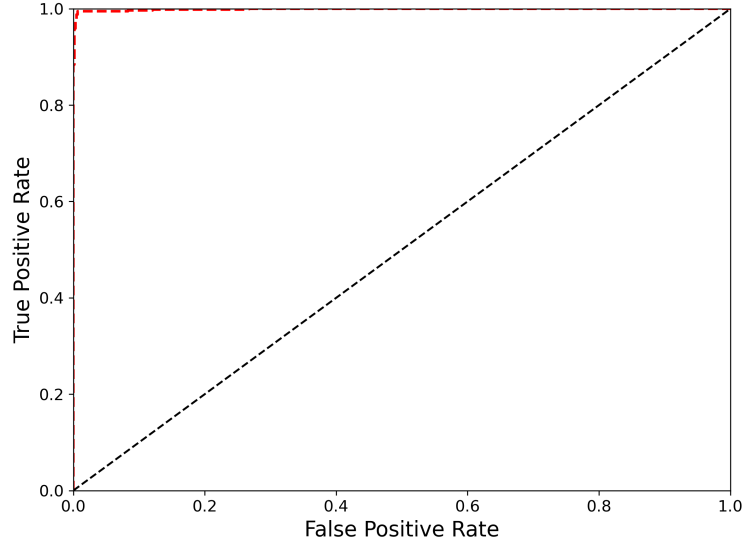


Figure 4.2: ROC curve of XG Boost classifier.

4.4.2 Evaluation of Features

We also assessed the effectiveness of each categories of feature set individually and also evaluated the performance of using the hybrid feature set. In this assessment, we have used the XG Boost classifier to classify the website. Figure 4.3 shows the classification performance measures using only the URL based features (UF2-UF15). URL-based features performs well with 97.79% true positive rate and 98.42% accuracy rate, as seen in the graph. Performance measures based on only hyperlink based features (HF1-HF10) are shown in figure 4.4. These features are the most important in correctly classifying phishing websites. Using only hyperlink features generates 84.67% accuracy and 96.93% true positive rate. In figure 4.5, we have presented the performance result of our proposed approach using the hybrid feature set that are obtained by combining all the features (UF2-UF15 and HF1-HF10). Our approach yields a high true positive rate (approximately 99%) with a low false positive rate (less than 0.5%). Both URL and hyperlink based features are useful for phishing website detection, but not enough to detect various types of phishing URLs. If hybrid feature set is used for phishing detection, we can be able to detect phishing attacks more accurately.

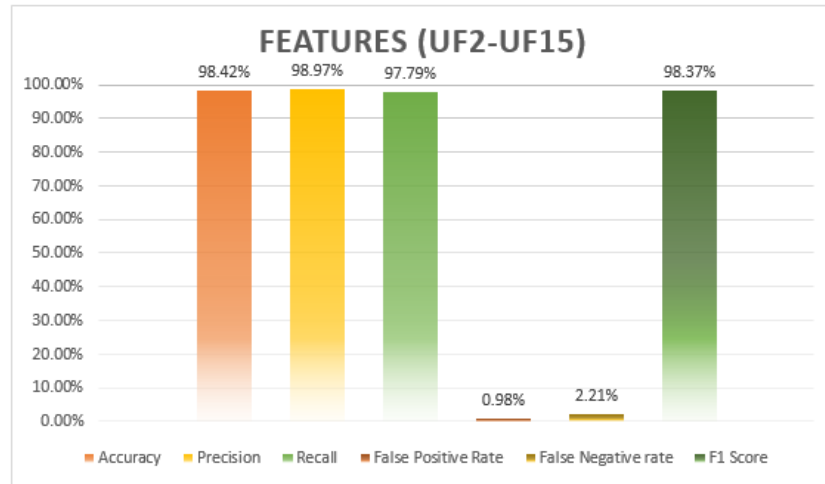


Figure 4.3: Performance measures based on URL Features.

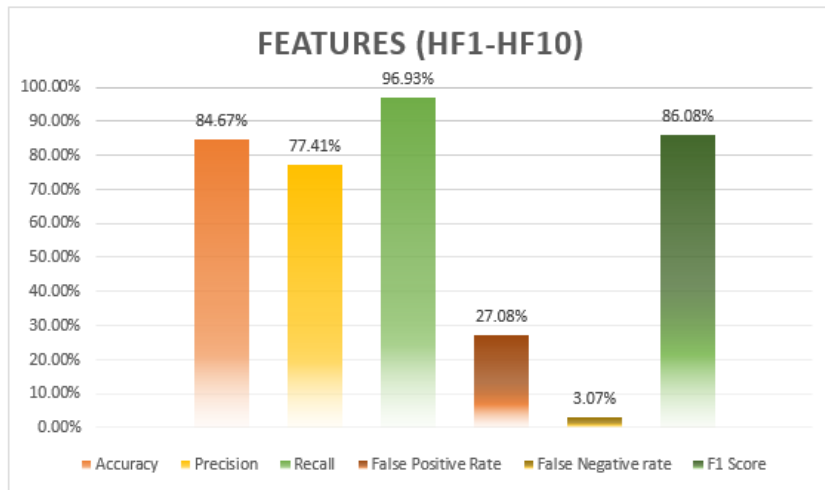


Figure 4.4: Performance measures based on Hyperlink features.

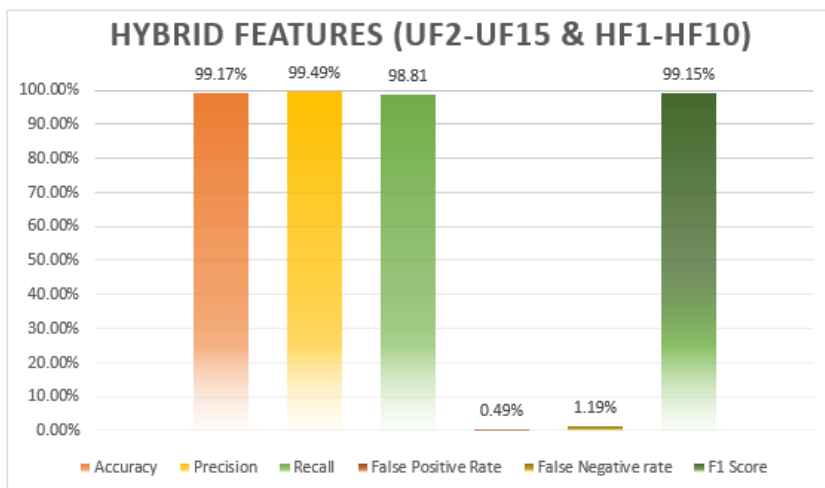


Figure 4.5: Overall performance measures using hybrid features.

4.4.3 Comparison with other Approach

We have compared the proposed mechanism to anti-phishing approaches that have been developed. In Table 4.1, we shows this comparison with the existing approaches. Our comparison is based on accuracy rate, True positive rate, language independent, search engine, third party independent and zero hour attack detection characteristics. The work of Rao et al.[1] gives higher accuracy rate compared to our work, but their work is dependent on mainly search engine and third party services. Only Rao et al.[1] and O. Sahingoz et al.[3] provide higher TPR than our proposed work. But they have overall lower performance than ours. Some of the work (Odeh et al. [17], Babagoli et al. [18]) can not detect the zero hour phishing attacks, which is a very important issue nowadays. Most of the approaches is not search engine independent, but search result shows the mostly popular sites on top of it and this feature extraction is also time consuming and complex. So, we didn't consider search engine based features in our approach. This characteristic of the phishing sites helps us to improve our performance.

Table 4.1: Comparison between various anti-phishing approaches based on performance

| Approach | Accur-
acy | TPR | Lan-
guage
Inde-
pendent | Search
engine
independ-
ent | Third
party
independ-
ent | Zero hour
attack
detection |
|------------------------------|---------------|-------|-----------------------------------|--------------------------------------|------------------------------------|----------------------------------|
| Rao et al.
[1] | 99.55 | 99.44 | Yes | No | No | Yes |
| A. Jain et
al. [2] | 98.42 | 98.39 | Yes | Yes | Yes | Yes |
| O.
Sahingoz
et al. [3] | 97.98 | 99.0 | Yes | Yes | Yes | Yes |
| Rao et al.
[16] | 94.26 | 93.31 | Yes | Yes | Yes | Yes |
| Odeh et
al. [17] | 98.9 | 98.6 | Yes | No | No | No |
| Babagoli
et al. [18] | 92.80 | 96.3 | Yes | No | No | No |
| Our
proposed
approach | 99.17 | 98.81 | Yes | Yes | Yes | Yes |

4.5 Conclusion

This chapter shows the performance evaluation result of proposed feature based anti-phishing strategy. Analyzing the performance we see that, our proposed hybrid feature set based methodology gives better phishing detecting capacity than other existing ones. In the next chapter, the conclusion is drawn on this thesis work.

Chapter 5

Conclusion

5.1 Conclusion

In this research, we have identified several efficient features for filtering phishing websites at the client-side where only URL based and Hyperlink based features are used. We have created a dataset of URLs by collecting from different sources and included various websites there to validate our proposed solution. The experimental results of our dataset proved that the proposed method is very effective as it has a 98.81% true positive rate and only 0.49% false positive rate. Our proposed method has more validity compared to other existing anti-phishing methods. The method depends entirely on the address bar and source code of a website. Therefore, our approach can be implemented at the client side also that will assist internet users in avoiding becoming victims of cyber criminals. We believe that if some more particular features can be added, then the accuracy of our method will be further improved. However, if other features are extracted from the third party, this will increase the complexity of the ongoing project. Our proposed hybrid feature set is totally dependent on the website's URL and source code, and it can only detect sites written in HTML code.

5.2 Future Work

Our future goal is to design a system that can detect non-HTML websites with high accuracy. Nowadays mobile devices have become the most popular everywhere and it seems to be a perfect target for malicious attacks like mobile phishing. So, mobile phishing will become the biggest threat in the future for us.

References

- [1] R. S. Rao and A. R. Pais, ‘Detection of phishing websites using an efficient feature-based machine learning framework,’ *Neural Computing and Applications*, vol. 31, no. 8, pp. 3851–3873, 2019 (cit. on pp. 1, 8, 33).
- [2] A. K. Jain and B. B. Gupta, ‘A machine learning based approach for phishing detection using hyperlinks information,’ *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 5, pp. 2015–2028, 2019 (cit. on pp. 1, 2, 8, 22, 24, 33).
- [3] Ö. K. Şahingöz, E. Buber, Ö. Demir and B. Diri, ‘Machine learning based phishing detection from uris,’ 2017 (cit. on pp. 1, 9, 33).
- [4] *Apwg q4 2020 report*, https://docs.apwg.org/reports/apwg_trends_report_q4_2020.pdf (accessed on 2 January 2021) (cit. on p. 2).
- [5] *Apwg h1 2014 report*, http://docs.apwg.org/reports/APWG_GlobalPhishingSurvey_1H2014.pdf (accessed on 2 October 2020) (cit. on p. 4).
- [6] A. K. Jain and B. B. Gupta, ‘Towards detection of phishing websites on client-side using machine learning based approach,’ *Telecommunication Systems*, vol. 68, no. 4, pp. 687–700, 2018 (cit. on p. 6).
- [7] S. Sheng, B. Magnien, P. Kumaraguru, A. Acquisti, L. F. Cranor, J. Hong and E. Nunge, ‘Anti-phishing phil: The design and evaluation of a game that teaches people not to fall for phish,’ in *Proceedings of the 3rd symposium on Usable privacy and security*, 2007, pp. 88–99 (cit. on p. 6).
- [8] Y. Wang, R. Agrawal and B.-Y. Choi, ‘Light weight anti-phishing with user whitelisting in a web browser,’ in *2008 IEEE Region 5 Conference*, IEEE, 2008, pp. 1–4 (cit. on p. 7).
- [9] W. Han, Y. Cao, E. Bertino and J. Yong, ‘Using automated individual white-list to protect web digital identities,’ *Expert Systems with Applications*, vol. 39, no. 15, pp. 11 861–11 869, 2012 (cit. on p. 7).
- [10] K. L. Chiew, E. H. Chang, W. K. Tiong *et al.*, ‘Utilisation of website logo for phishing detection,’ *Computers & Security*, vol. 54, pp. 16–26, 2015 (cit. on p. 7).
- [11] A. P. Rosiello, E. Kirda, F. Ferrandi *et al.*, ‘A layout-similarity-based approach for detecting phishing pages,’ in *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops SecureComm 2007*, IEEE, 2007, pp. 454–463 (cit. on p. 7).

- [12] K. L. Chiew, J. S.-F. Choo, S. N. Sze and K. S. Yong, ‘Leverage website favicon to detect phishing websites,’ *Security and Communication Networks*, vol. 2018, 2018 (cit. on p. 7).
- [13] M. Felegyhazi, C. Kreibich and V. Paxson, ‘On the potential of proactive domain blacklisting.,’ *LEET*, vol. 10, pp. 6–6, 2010 (cit. on p. 7).
- [14] A. K. Jain and B. B. Gupta, ‘Phishing detection: Analysis of visual similarity based approaches,’ *Security and Communication Networks*, vol. 2017, 2017 (cit. on p. 7).
- [15] Y. Huang, J. Qin and W. Wen, ‘Phishing url detection via capsule-based neural network,’ in *2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, IEEE, 2019, pp. 22–26 (cit. on p. 9).
- [16] R. S. Rao, T. Vaishnavi and A. R. Pais, ‘Catchphish: Detection of phishing websites by inspecting urls,’ *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 2, pp. 813–825, 2020 (cit. on pp. 9, 33).
- [17] A. Odeh, I. Keshta and E. Abdelfattah, ‘Phiboost-a novel phishing detection model using adaptive boosting approach,’ *Jordanian Journal of Computers and Information Technology (JJCIT)*, vol. 7, no. 01, 2021 (cit. on pp. 9, 33).
- [18] M. Babagoli, M. P. Aghababa and V. Solouk, ‘Heuristic nonlinear regression strategy for detecting phishing websites,’ *Soft Computing*, vol. 23, no. 12, pp. 4315–4327, 2019 (cit. on pp. 9, 33).
- [19] R. M. Mohammad, F. Thabtah and L. McCluskey, ‘Predicting phishing websites based on self-structuring neural network,’ *Neural Computing and Applications*, vol. 25, no. 2, pp. 443–458, 2014 (cit. on p. 9).
- [20] F. Feng, Q. Zhou, Z. Shen, X. Yang, L. Han and J. Wang, ‘The application of a novel neural network in the detection of phishing websites,’ *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–15, 2018 (cit. on p. 10).
- [21] *Phishtank opensource platform*, <http://phishtank.org/> (accessed on 2 October 2020) (cit. on p. 11).
- [22] *University of new brunswick,url dataset*, <https://www.unb.ca/cic/datasets/url-2016.html> (accessed on 2 October 2020) (cit. on p. 12).
- [23] *Alexa - find top websites*, <https://www.alexa.com/> (accessed on 2 October 2020) (cit. on p. 12).
- [24] G.-G. Geng, X.-T. Yang, W. Wang and C.-J. Meng, ‘A taxonomy of hyperlink hiding techniques,’ in *Asia-Pacific web conference*, Springer, 2014, pp. 165–176 (cit. on p. 22).