Detection of Phishing Websites Hosted on Free Web Hosting Domains Using Machine Learning

A project report submitted in fulfillment of the requirements

for the award of

Degree of Bachelor of Technology in Computer Science and Engineering

by

Rohit Lohar (2019UCP1373)

Sneh Aashish Gupta (2019UCP1900)

Divya Rathore (2019UCP1370)

Under the supervision of

Dr. Meenakshi Tripathi, Associate Professor

MNIT, Jaipur

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY

MAY 2023

CERTIFICATE

We ,

ROHIT LOHAR (2019UCP1373)

SNEH AASHISH GUPTA (2019UCP1900)

DIVYA RATHORE (2019UCP1370)

Declare that this thesis titled, **"Detection of Phishing Websites Hosted on Free Web Hosting Domains Using Machine Learning "** and the work presented in it are our own. We confirm that :

- This project work was done wholly or mainly while in candidacy for a B.Tech. degree in the department of Computer Science and Engineering at Malaviya National Institute of Technology Jaipur (MNIT).
- Where any part of this report has previously been submitted for a degree or any other qualification at MNIT or any other institution, this has been clearly stated. Where we have consulted the published work of others, and this is always clearly attributed. Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this project is entirely our own work.
- We have acknowledged all the main sources of help.

Signature

Date

Dr. Meenakshi Tripathi
Associate Professor
Department of Computer Science and Engineering
Malaviya National Institute of Technology Jaipur

thesis

ORIGINALITY REPORT

11%	7%	3%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.researchgate.net Internet Source	1%
2	deepai.org Internet Source	1%
3	www.mdpi.com Internet Source	<1%
4	Submitted to Coventry University Student Paper	<1%
5	Tanmoy Sarkar Pias, Yiqi Su, Xuxin Tang, Haohui Wang, Danfeng (Daphne) Yao. "Undersampling for Fairness: Achieving More Equitable Predictions in Diabetes and Prediabetes", Cold Spring Harbor Laboratory, 2023 Publication	<1%
6	Submitted to University of Salford Student Paper	<1%
7	Submitted to University of North Texas Student Paper	<1%

DECLARATION

This is to certify that this major project titled ”Detection of Phishing Websites Hosted on Free Web Hosting Domains Using Machine Learning,” submitted by Rohit Lohar,Sneh Aashish Gupta and Divya Rathore in partial fulfillment of the requirements for the award of the degree Bachelor of Technology in the Department of Computer Science and Engineering at Malaviya National Institute of Technology has a record of bona fide major project work. Carried out by him under my supervision. The contents of this major project, in whole or in parts, has not been submitted to any other institute or university for the award of any degree or diploma.

The content of this report, in whole or in parts, have not been reproduced from any existing work of any other person and has not been submitted anywhere else at any time.

Signature

Date

Dr. Meenakshi Tripathi
Associate Professor
Department of Computer Science and Engineering
Malaviya National Institute of Technology Jaipur

ACKNOWLEDGEMENT

We would like to express our gratitude to all those who helped us realize and nourish our work.
We express our sincere regards and gratitude to Dr. Meenakshi Tripathi, Associate Professor, MNIT Jaipur, for her inspiration, expert guidance, morale-boosting, continuous encouragement, and appreciation, which are vital factors in the successful completion of our major project work. We are grateful for this support and for allowing us to explore several opportunities. We humbly acknowledge our deep gratitude towards our guide and mentor. Throughout our project's stages, she helped us keep track of the right research direction, patiently discussed, and seriously revised all our work. We would also like to thank Dr. Hossein Shirazi, Management Information Systems Department, San Diego State University, for his guidance in our progress, helping us stay focused on problems and providing new directions to approach them. We would be thankful to them for all our lives, for guiding us on various matters, always motivating us, and boosting our confidence, which helped us shape our lives.
Lastly, we would like to express our heart-felt gratitude to our parents and our family. Thank you for giving us so much support.

ROHIT LOHAR	**2019UCP1373**
SNEH AASHISH GUPTA	**2019UCP1900**
DIVYA RATHORE	**2019UCP1370**

ABSTRACT

Even though phishing attacks have changed over time to use a variety of ways to hide themselves from common detection methods, using these methods often takes a lot of technical knowledge and planning on the part of the offender. We identify a family of phishing websites hosted over free web hosting domains (FHDs), which can be created and maintained at scale with very little effort, while also effectively evading prevalent anti-phishing detection and resisting website takedowns. A large-scale study of these websites shows that phishing websites hosted on FHDs stay online longer than normal phishing URLs. On average, they have less coverage from anti-phishing blocklists than regular phishing attacks. Their coverage time is also slower, and anti-phishing tools only catch half as many of them. From August 2022 to April 2023, more than 5,000 of these FHD URLs were observed on Phishtank.com and Openphish.com by 50 different FHD providers. Looking at various literature surveys, not much work has been done related to FHD URLs with trustworthy features and higher detection accuracy.

This thesis finds a family of phishing websites that are held on free web hosting domains (FHDs). It deals with identifyning effective features from the FHD URLs and creation of up-to-date dataset. It also deals with machine learning algorithms for the detection of phishing URLs hosted on FHDs by narrowing down the best machine learning algorithm by comparing the accuracy rate, false positive rate, and false negative rate of various algorithms.

CONTENTS

Certificate **i**

Declaration **iii**

Acknowledgement **iv**

Abstract **v**

Contents **vi**

List of Figures **viii**

List of Tables **ix**

1 INTRODUCTION **1**

1.1 Motivation . . . 2

1.2 Problem statement . . . 4

1.3 Objectives . . . 5

1.4 Key Contributions . . . 5

2 RELATED WORKS **7**

2.1 Learning algorithms for phishing detection . . . 12

2.1.1 Logistic Regression . . . 13

2.1.2 Decision Tree . . . 14

2.1.3 Random Forests . . . 16

2.1.4 Support Vector Machine(SVM) . . . 17

2.1.5 Multilayer Perceptron (MLP) . . . 18

2.1.6 XGBoost Classifier . . . 19

2.2 Performance metrics for Classification . . . 20

3 Proposed Methodology **24**

3.1 URL Collection . . . 24

3.2 Features Extraction . . . 25

3.2.1 Address Bar Based Features 25

3.2.2 Domain Based Features 28

3.2.3 HTML and JavaScript based Features 29

3.3 Dataset Creation 35

4 Performance Evaluation **39**

4.1 Experimental Setup 39

4.2 Training and testing machine learning models on phishing dataset 41

4.3 Training and testing machine learning models on FHD dataset 44

5 Conclusions and Future Works **47**

5.1 Conclusions 47

5.2 Future Works 47

References **48**

List of Figures

1.1 Number of brands affected by phishing attacks in following year 2
1.2 Number of unique phishing websites detected in following year 3
1.3 Targeted industries by hackers 4
1.4 Life Span of phishing URLs in subsequent years 5

2.1 Data accuracy 7
2.2 Data accuracy 9
2.3 Data accuracy 9
2.4 The FreePhish framework 10
2.5 High level architecture of the collaborative learning system 12
2.6 Sigmoid Curve 13
2.7 Decision Tree 15
2.8 Random Forest Trees 16
2.9 SVC Diagram 17
2.10 MLP Diagram 18
2.11 Confusion Matrix 21
2.12 AUC-ROC Curve 23

3.1 Flowchart of phishing detection 24
3.2 Count of Indexed Pages 33
3.3 VirusTotal UI 38

4.1 Feature Importance of phishing dataset on random forest classifier 42
4.2 Confusion Matrix and AUC-ROC curve on phishing dataset 42
4.3 Performance metrics on phishing dataset 43
4.4 Feature Importance of FHD dataset on multilayer perceptron classifier . . . 45
4.5 Confusion Matrix and AUC-ROC curve on FHD dataset 45
4.6 Performance metrics on FHD dataset 46

List of Tables

3.1 Address Bar based Features 34
3.2 Domain based Features 34
3.3 HTML and Javascript based Features 35
3.4 Free Hosting Domains by Roy et al.[1] 36
3.5 Some more Free Hosting Domains added during study 37
3.6 Number of URL collected in each catagory 38

4.1 Performance of ML Models on phishing dataset 41
4.2 Performance Metrics on phishing dataset 43
4.3 Comparison of Existing Methodology with Proposed Methodology 43
4.4 Performance of ML Models on FHD dataset 44
4.5 Performance Metrics on FHD dataset 46
4.6 Comparison of Existing Methodology with Proposed Methodology 46

Chapter 1
INTRODUCTION

Phishing is a type of cyber attack that involves using fraudulent emails, text messages, or other forms of electronic communication to trick individuals into providing sensitive information, such as passwords, credit card numbers, or other personal data.
Phishing websites are fake websites that are designed to look like legitimate websites in order to steal personal or financial information from unsuspecting users. These websites are usually created by cybercriminals who send phishing emails or messages to potential victims, luring them to click on a link that takes them to the fake website. Phishing websites are designed to look like popular websites such as banks, social media platforms, online retailers, and email providers. They often use logos, branding, and other visual elements to create a convincing imitation of the legitimate site. Once the user is on the fake website, they may be prompted to enter their login credentials, personal information, or financial details, which are then captured by the attacker.
A large-scale study of these attacks shows that phishing web- sites hosted on FHDs stay online at least 1.5 times longer than normal phishing URLs. On average, they have 1.7 times less coverage from anti-phishing blocklists than regular phishing attacks. Their coverage time is also 3.8 times slower, and anti-phishing tools only catch half as many of them. Also, the hosting site took an average of 12.2 hours to get rid of only 23.6% of FHD URLs a week after they first appeared [1].
To protect against these phishing websites,features have to be collected based on these websites (both phishing and legitimate types) and trained on various machine learning models to create a reliable and steady detection system to classify these URLs as phishing and legitimate ones. The solutions to the problem scenario also cover the increasing complexity of websites, which allows attackers to try new features and techniques to improve the quality of phishing websites.

1.1 Motivation

It is important to have reliable detection systems to check for phishing websites because phishing attacks have become increasingly common and sophisticated in recent years. According to a report by the Anti-Phishing Working Group, there were more than 222,000 unique phishing attacks in the first quarter of 2021 alone.

Phishing attacks can have severe consequences for individuals and organizations, including identity theft, financial loss, and reputational damage. By detecting and blocking phishing websites, detection systems can help prevent these negative outcomes and protect individuals and organizations from harm.

Furthermore, as more people conduct their personal and professional lives online, the risk of falling victim to phishing attacks has increased. Phishing attacks have also become more targeted, with attackers using information obtained from social media and other sources to personalize their attacks and make them more convincing. Reliable detection systems can help identify and block these targeted attacks.

Some of the facts about phishing attacks that should be known in 2023[2]:-

- The F5 Labs Phishing and Fraud Report of 2020 says that targeted brand names are used by 55% of phishing websites to make it easy to get private information as shown in Figure 1.1.

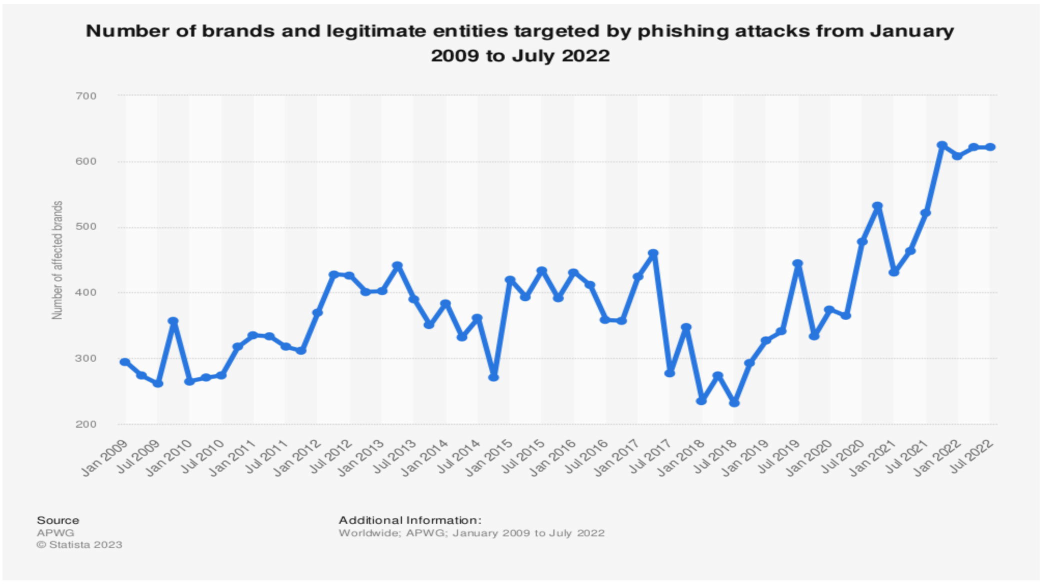


Figure 1.1: Number of brands affected by phishing attacks in following year

- The top four most commonly spoofed tech companies are Amazon (13%), Google (8%), Facebook (9%), and Apple (2%) in that order.

- One of the main reasons for data breaches is phishing. The average cost of a data breach rose from \$4.24m in 2021 to \$4.35m in 2022, according to IBM's 2022 Cost of Data Breach Report [3].

- According to Verizon's analysis from 2022, phishing was implicated in 36% of all data breaches. By 2022, one ransomware or phishing assault was predicted to happen every 11 seconds [4].

- The number of phishing attacks reached an all-time high in 2022, according to the APWG's Phishing Activity Trends Report for 3rd quarter of 2022. This was the worst quarter ever, with more than 1,270,000 strikes in 3rd quarter alone [5] as shown in Figure 1.2.

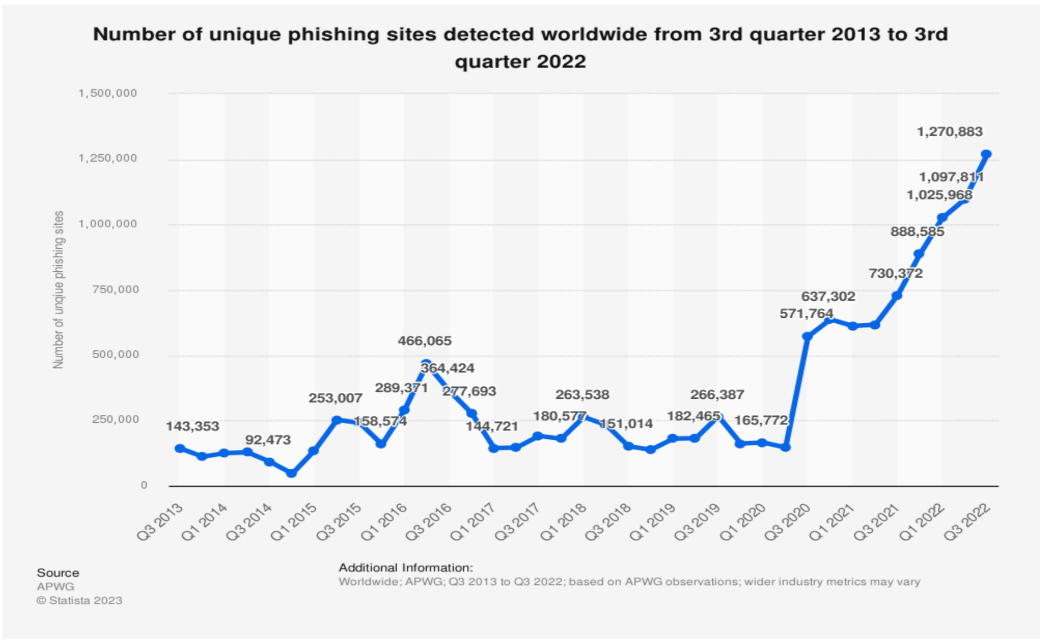


Figure 1.2: Number of unique phishing websites detected in following year

- There are many different types of targets that attackers choose as shown in Figure 1.3.

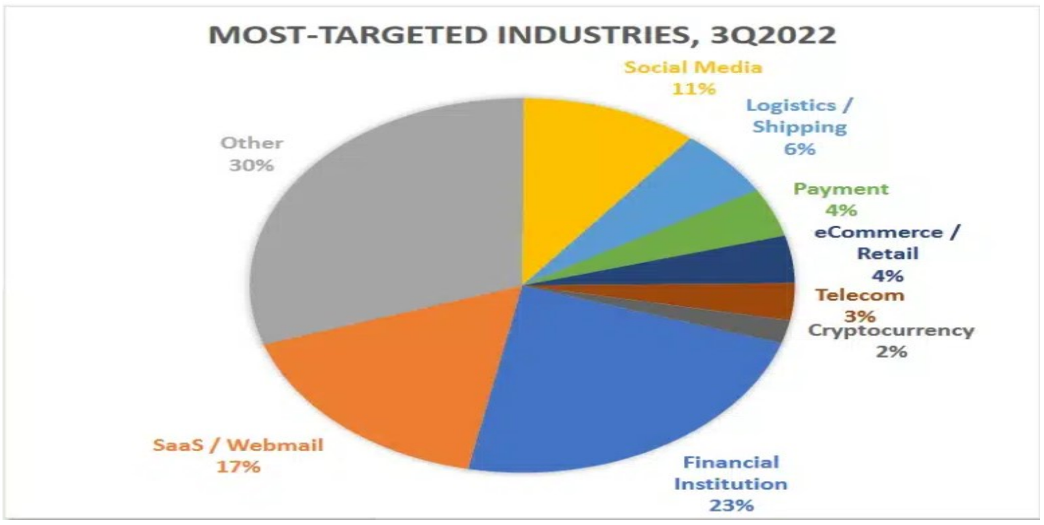


Figure 1.3: Targeted industries by hackers

It is important to note that these statistics are constantly changing as attackers continue to develop new techniques and tactics to evade detection and steal sensitive information. As such, it is critical to stay vigilant and take proactive measures to protect against phishing attacks.

1.2 Problem statement

Phishing detection is a serious problem. It refers to the weakness of the user, which makes them vulnerable to such attacks. There is not only one solution to minimize the vulnerabilities effectively; thus, multiple techniques are implemented. Phishing detection plays an important role in avoiding the loss of personal data by users. We identify a family of phishing websites hosted over free web hosting domains (FHDs), which can be created and maintained at scale with very little effort while also effectively evading prevalent anti-phishing detection and resisting website takedowns. Phishing websites usually only stay online for a short time. This makes it harder to get enough information since a single person or group collecting information is likely to miss some of the fake websites. In 2016, about 84% of scam sites were only around for 24 hours. The average amount of time a fake website was up and running was about 15 hours. In 2018, it was down to less than 50 minutes [6]. This shows that scam sites tend to be around for less and less time as time goes on. So, it is hard to get a large enough amount of training data for these kinds of websites. The life span of phishing URLs is shown in Figure 1.4 for subsequent years. Hence, there is a need for an up-to-date dataset with trustworthy features.

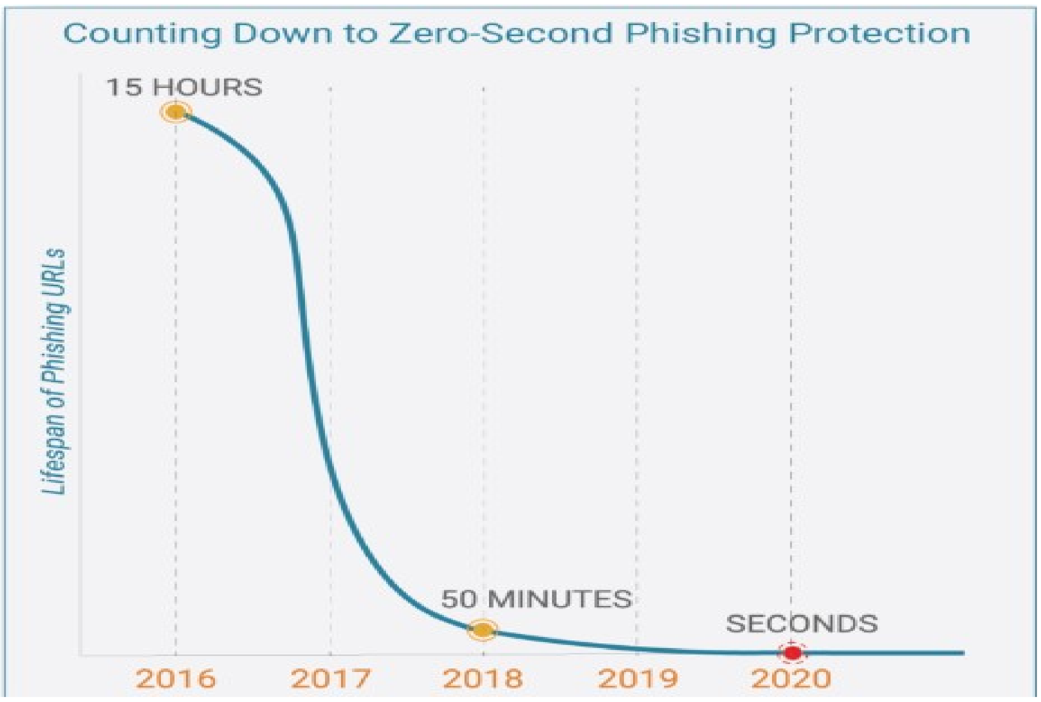


Figure 1.4: Life Span of phishing URLs in subsequent years

1.3 Objectives

Detection of phishing websites hosted on free web hosting domains (FHDs) using machine learning techniques

- to identify phishing websites hosted on free web hosting domains.
- to come up with trustworthy features and creation of the up-to-date dataset.
- to develop machine learning models to precisely detect phishing websites on FHD.

1.4 Key Contributions

Some key contributions by us are as follows :

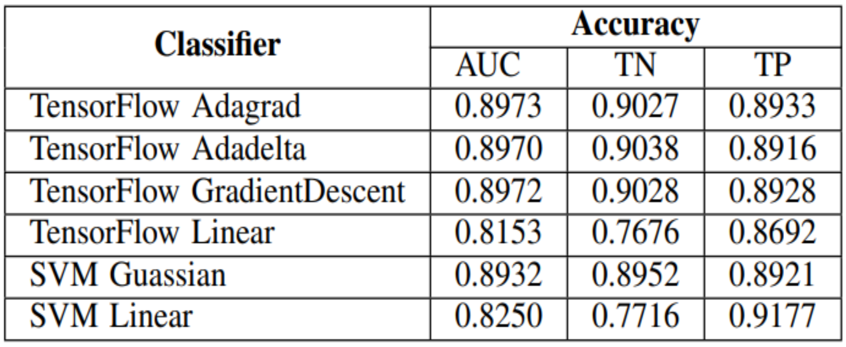
- Identifyning effective features from the FHD URLs and creation of up-to-date dataset.
- Highlighting the gaps in the current anti-phishing environment in detecting FHD URLs with trustworthy features and higher detection accuracy. We proposed a set of effective features and ML algorithms to precisely detect phishing websites on FDH.

In this chapter, we described phishing, its motivation for effective detection, its objectives, and its contributions. In the next chapter, we will be talking related works about phishing detection, different learning algorithms for phishing detection, and performance metrics in classification.

Chapter 2
RELATED WORKS

This chapter covers the concepts and theories required for this project. The outcomes of this chapter will be the characteristics and features based on which phishing and legitimate websites will be differentiated. How are these features implemented by attackers to mimic the legitimate website of a reputable organization, such as a bank or other financial institution? This is a ploy to get you to reveal sensitive information, including your login passwords or financial details. The problem arises, however, when a user willingly submits personal information to a website whose owner is evil and then utilizes that information to exploit the user and his accounts. After that, learn how different machine learning models work and classify our objective task.

Shirazi et al. [7] made the FreshPhish framework because there isn't an open-source framework that measures a website's features. Also, they made a set of up-to-date facts that other researchers can use. They used 6,000 legitimate websites and 6,000 phishing websites to make their Fresh-Phish dataset. They took 10 important features like URL length, Using shortening service, Prefix and suffix, Prefix and suffix, HTTPS token, Request URL, Redirect page, OnMouseOver, PopUp widnow and Google index. They then used this dataset to train their classifier. They also looked at the FreshPhish data set with a TensorFlow neural network, a TensorFlow linear classifier, an SVM with a Gaussian kernel, and an SVM with a linear kernel. The results of the following algorithm were given in Figure 2.1



Classifier	Accuracy		
	AUC	TN	TP
TensorFlow Adagrad	0.8973	0.9027	0.8933
TensorFlow Adadelta	0.8970	0.9038	0.8916
TensorFlow GradientDescent	0.8972	0.9028	0.8928
TensorFlow Linear	0.8153	0.7676	0.8692
SVM Guassian	0.8932	0.8952	0.8921
SVM Linear	0.8250	0.7716	0.9177

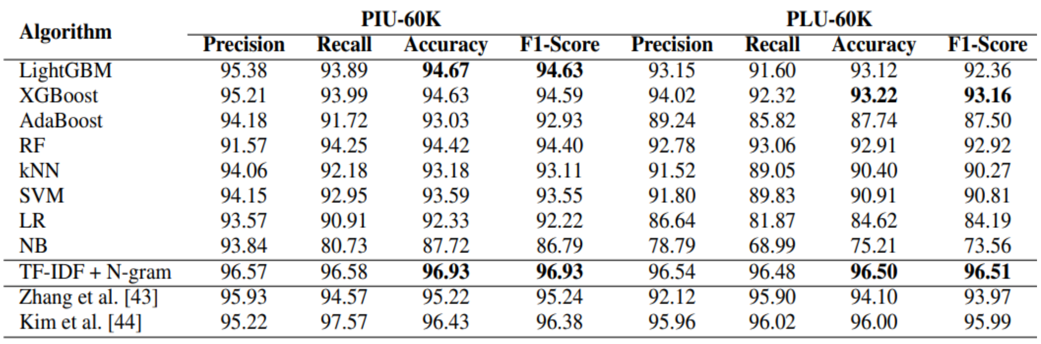
Figure 2.1: Data accuracy

They found that the TensorFlow implementations took a lot longer to train and weren't much better than the SVM in terms of accuracy. Using a Gaussian kernel SVM, they got a 90% success rate for the fresh phishing test data.

In [8], they extend their earlier dataset, called PILU-60K (Phishing Index Login URL), from 60K to 90K URLs that were evenly split between three groups: phishing, the real homepage, and the real login. They make this larger dataset, PILU-90K, available to the public for study. They used PILU-90K to construct and assess three URL phishing detection pipelines:

- They train eight supervised machine learning classifiers using the 38 handcrafted feature descriptors supplied by Sahingoz et al.
- a Convolutional Neural Network (CNN) that also operates at the character level
- a Logistic Regression (LR) algorithm that uses character N-grams as inputs.

They provided empirical evidence that a machine trained on the URLs of phishing and authentic homepages has difficulty classifying login URLs. In this configuration, handcrafted features were extracted and several classifiers were evaluated, as described. Each model was trained and evaluated with respect to every subset of the PILU-90K dataset. The effectiveness of each classifier is depicted in Figure. XGBoost, LightGBM, and RF outperform the other classifiers on both subsets, achieving 93.22 percent, 93.12 percent, and 92.91% accuracy on PLU60K, respectively. While the PIU-60K sample subset achieved an accuracy of 94.63, 94.67, and 94.42%, respectively as shown in Figure 2.2. As a result of their efforts, researchers now have access to an improved dataset (PILU-90K) on which to train and evaluate their methods. Authentic login URLs, the most realistic case for phishing detection, are included in this dataset. Using deep learning and machine learning solutions trained on phishing and real home URLs, they investigated a number of URL-based detection algorithms. Their method's key benefit is its accuracy in identifying legitimate URLs for classification. With an accuracy of 96.50%, TF-IDF paired with the N-gram and LR algorithms performed best of the models tested.



Algorithm	PIU-60K				PLU-60K			
	Precision	Recall	Accuracy	F1-Score	Precision	Recall	Accuracy	F1-Score
LightGBM	95.38	93.89	**94.67**	**94.63**	93.15	91.60	93.12	92.36
XGBoost	95.21	93.99	94.63	94.59	94.02	92.32	**93.22**	**93.16**
AdaBoost	94.18	91.72	93.03	92.93	89.24	85.82	87.74	87.50
RF	91.57	94.25	94.42	94.40	92.78	93.06	92.91	92.92
kNN	94.06	92.18	93.18	93.11	91.52	89.05	90.40	90.27
SVM	94.15	92.95	93.59	93.55	91.80	89.83	90.91	90.81
LR	93.57	90.91	92.33	92.22	86.64	81.87	84.62	84.19
NB	93.84	80.73	87.72	86.79	78.79	68.99	75.21	73.56
TF-IDF + N-gram	96.57	96.58	**96.93**	**96.93**	96.54	96.48	**96.50**	**96.51**
Zhang et al. [43]	95.93	94.57	95.22	95.24	92.12	95.90	94.10	93.97
Kim et al. [44]	95.22	97.57	96.43	96.38	95.96	96.02	96.00	95.99

Figure 2.2: Data accuracy

In [9], the software's objective is to categorize phishing attacks, which are a subset of cyber dangers. For this, they turn to the Extreme Learning Machine. UCI's online resources were mined for data for this investigation. There are 30 input attributes that determine 1 output attribute in this data set. There are three possible values for input attributes: 1, 0, and -1. The values 1 and -1 are accepted for the output attribute. In this research, they evaluate the developed system's efficacy using a 10-fold cross-validation test. The study found that 95.05% was the average classification accuracy and 95.93% was the maximum accuracy achieved. Their proposed architecture are given in Figure 2.3

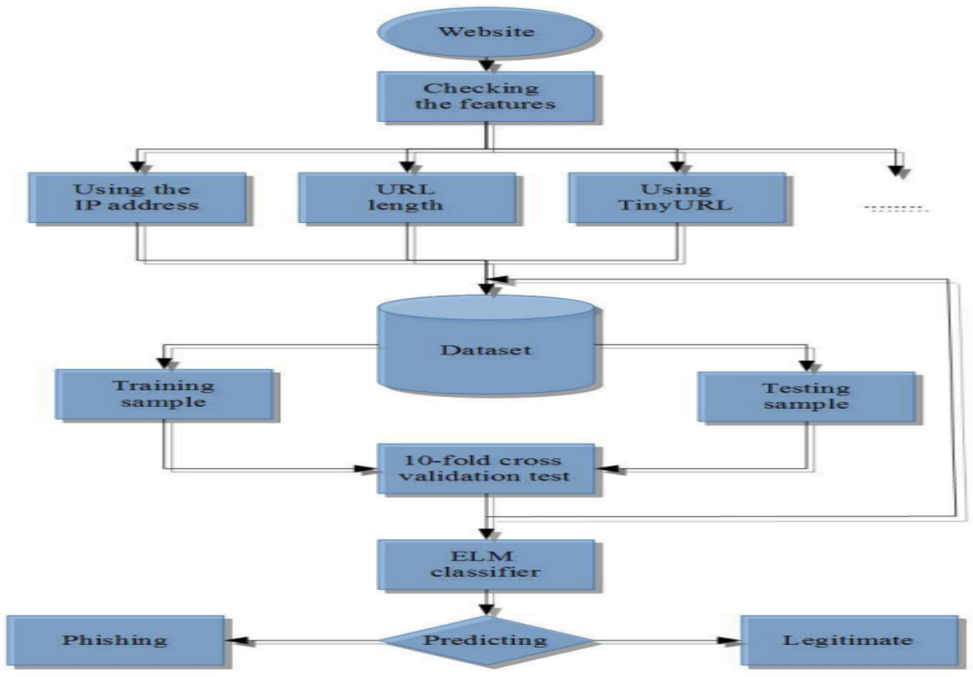


Figure 2.3: Data accuracy

Roy et al. [1] found and categorized a new group of phishing sites that take advantage of free hosting services. They discovered that attackers typically launch such attacks on social networking websites by taking advantage of features inherent to these domains. They conducted a measuring research and discovered that, in comparison to phishing websites housed on ordinary (self-hosted) domains, these types of attacks are substantially more successful at escaping detection by anti-phishing software. In addition, they discovered that adversaries employ FHDs to propagate malware and to host complex phishing operations. Their research led them to develop a system called FreePhish as shown in Fgure 2.4, which can identify and report FHD-based phishing assaults when they occur in the wild. Over the course of two weeks, they found over 1,200 unique FHD-hosted phishing URLs and reported them to the relevant FHD provider, greatly increasing the frequency and speed with which these sites were taken down. In order to prevent users from FHD-based phishing assaults, they offer FreePhish as a free web addon for Chrome-based web browsers. Their research will assist bring greater attention to the issue of FHD-hosted phishing assaults, and their FreePhish framework will aid in the efficient removal of such attacks from domain registrars and social networking platforms alike. Using their model's Random Forest Classifier, they were able to achieve an accuracy of 97.30% in their classifications.

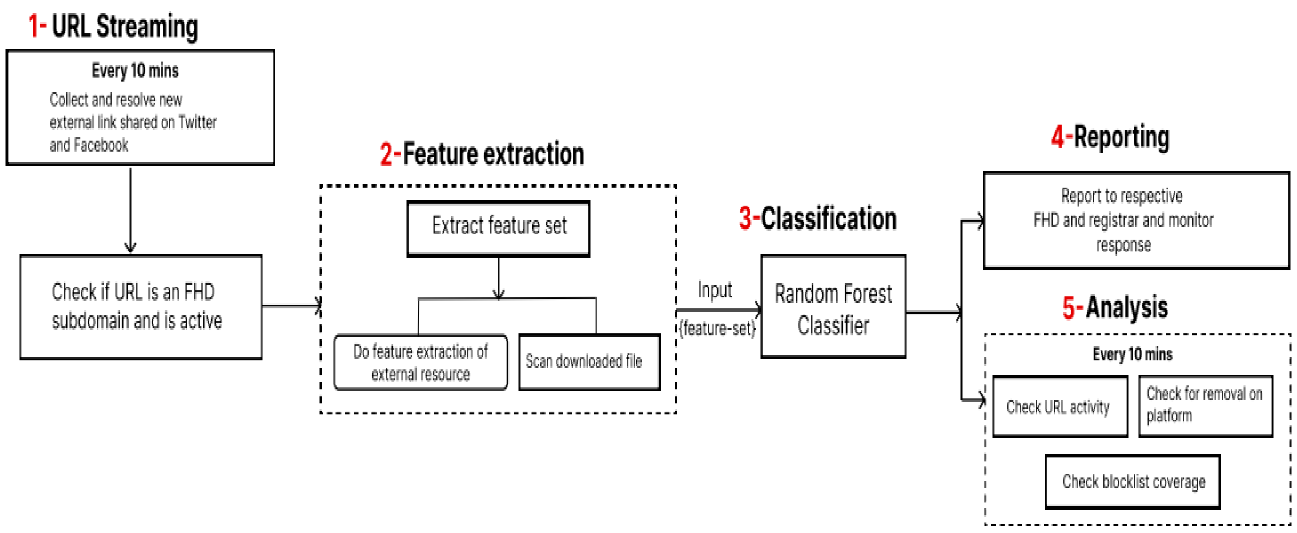

1- URL Streaming
Every 10 mins
Collect and resolve new external link shared on Twitter and Facebook
Check if URL is an FHD subdomain and is active
2-Feature extraction
Extract feature set
Do feature extraction of external resource
Scan downloaded file
Input
{feature-set}
3-Classification
Random Forest Classifier
4-Reporting
Report to respective FHD and registrar and monitor response
5-Analysis
Every 10 mins
Check URL activity
Check for removal on platform
Check blocklist coverage

Figure 2.4: The FreePhish framework

Researcher struggled to find reliable training datasets. There are several data mining articles regarding predicting phishing websites, but no publicly available training dataset. In [10], they have found some effective features that are powerful in detecting phishing websites. They found out 30 effective features, which are grouped into three categories: address bar-based, domain-based, and HTML and Javascript-based. Some features implemented in our work were originally defined by Mohammad et al.

In [11], Figure 3.1 depicts the primary components and flow of our collaborative training system. The framework implies that N participants are involved in the training procedure. Each participant is either a local consumer or participant. All participants concur in advance on a standardized neural network architecture. The parameter server is responsible for administering a list of global parameters that, in essence, represent the shared model trained collaboratively by all local clients. Their model achieved a high TPR of 97.78% in a 4-client environment. The average TPR for a 4-client system is recorded as 95.41% (for 6 features), 96.59% (for 7 features), 96.75% (for 9 features), and 97.22% (for 11 features). Along with weight gradients, they also performed another variant of all the experiments where bias values were also shared among participants. This proved to be a significant method to tackle the imbalanced dataset. Furthermore, their model is able to adapt to new phishing features over the years. For this, they introduced the concept of a null feature vector that reserves some space for new potential features. With this, their model can be trained on new features by just making changes in the feature vector and not in the model's architecture itself. Also, during the data collection phase of the project, they found some new features for phishing websites that make their model robust for newer website phishing attacks.

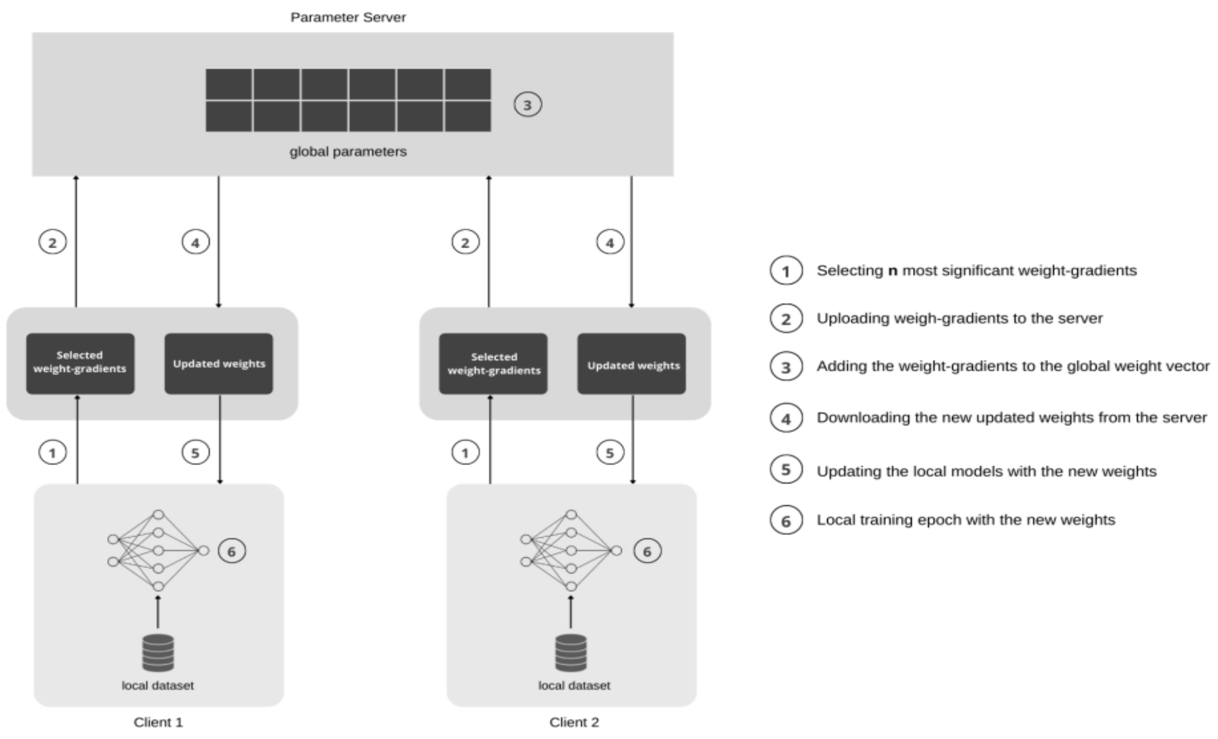

Parameter Server
3
global parameters
2
4
2
4
Selected weight-gradients
Updated weights
Selected weight-gradients
Updated weights
1
5
1
5
6
6
local dataset
local dataset
Client 1
Client 2
1 Selecting n most significant weight-gradients
2 Uploading weigh-gradients to the server
3 Adding the weight-gradients to the global weight vector
4 Downloading the new updated weights from the server
5 Updating the local models with the new weights
6 Local training epoch with the new weights

Figure 2.5: High level architecture of the collaborative learning system

So looking at this literature survey, there is still scope for improving phishing detection of FHD-based URLs because not much work has been done on this type of URL. Some authors have performed their experiments with a smaller number of features, which is not trustworthy to cover every characteristic aspect of phishing URLs. Either some authors have performed their experiments on a lesser number of URLs in their dataset or they have achieved lesser accuracy.

2.1 Learning algorithms for phishing detection

The problem of finding phishing websites can be thought of as a binary classification job, where the goal is to put a website into one of two categories: phishing or legitimate. To find phishing websites, different features can be taken from the text and metadata of the site. Once the features are extracted, a machine learning model can be taught on a labeled dataset to tell if a website is phishing or not. Logical regression, decision trees, random forests, support vector machines (SVMs), and neural networks are all methods that are often used for binary classification. These algorithms are discussed below.

2.1.1 Logistic Regression

Logistic regression is a popular algorithm for machine learning that is used for jobs that can be broken down into two groups. It uses the probability of y given x to model the link between a binary dependent variable (y) and one or more independent variables (x).The logistic function, which is also called the sigmoid function, is used by the logistic regression model to map the input values to a chance between 0 and 1 [12]. This is what the sigmoid function means:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.1}$$

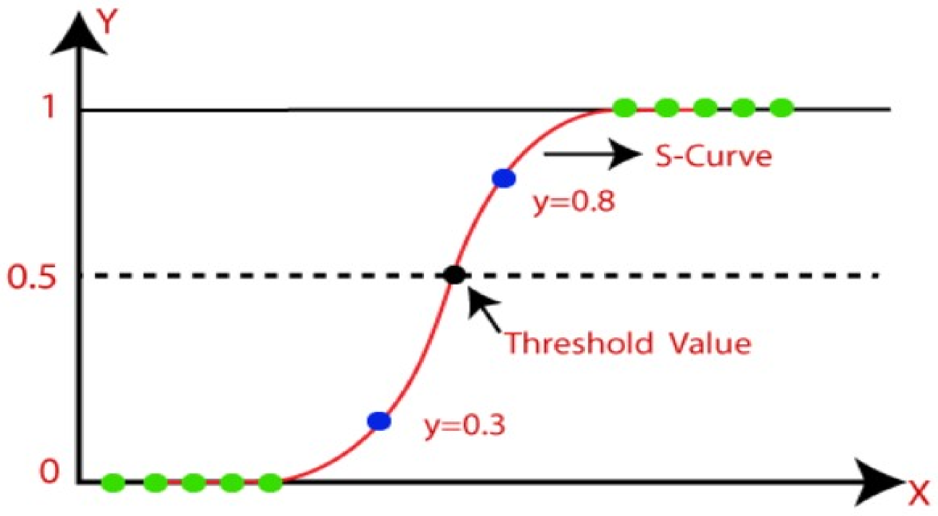


Figure 2.6: Sigmoid Curve

where z is a linear combination of the traits and weights that were given as input:

$$z = w\_0 + w\_1x\_1 + w\_2x\_2 + ... + w\_n * x\_n \tag{2.2}$$

where w_0 is the intercept term, w_1–w_n are the weights for the input features x_1–x_n, and n is the amount of input features.

A training set of data is used by the logistic regression model to figure out the best numbers for the weights w_0 to w_n that minimize a cost function. The binary cross-entropy loss function is the most popular cost function. This function is defined as:

$$L(y, y_h a) = -(y * log(y_h at) + (1 - y) * log(1 - y_h at)) \tag{2.3}$$

where y is the real label (0 or 1), y_hat is the predicted chance, and log is the natural logarithm. The goal of training the logistic regression model is to find the values of w_0 to w_n that make the average value of the loss function lowest across all training samples.
Once the model has been trained, it can be used to make guesses about new data by computing the output of the sigmoid function for each input feature set. If the output is more than a certain threshold, which is usually 0.5, the expected label is 1, and if it is less than that threshold, it is 0.
Logistic regression is used a lot for things like credit scores, finding fraud, and making medical diagnoses, among other things. It is a simple algorithm that works well for jobs that can be split into two groups. It is also easy to understand and explain.

2.1.2 Decision Tree

A prominent supervised machine learning technique for classification and regression is decision trees.A decision tree is a flowchart-like structure with internal nodes representing features, branches representing decision rules, and leaf nodes representing decision outcomes. Recursively partitioning the dataset by features, the algorithm learns the decision tree until it reaches the leaf node that gives the classification or regression result[13].

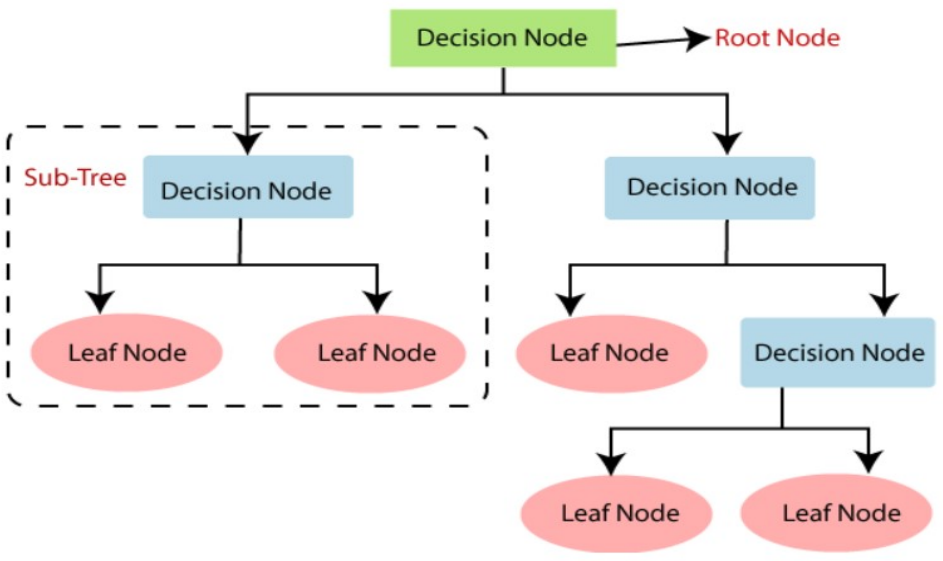


Figure 2.7: Decision Tree

Decision trees' major advantages are:

1. Easy to understand: Decision trees graphically reflect the decision-making process, making it simple to understand and explain.
2. Decision trees can accommodate category and numerical data, making them suitable for mixed datasets.
3. Non-parametric: Decision trees don't need data distribution or variable interrelationship assumptions.
4. Decision trees can handle high-dimensional data, making them suited for complicated datasets.
5. Decision trees can manage missing data, making them helpful for partial datasets.

However, decision trees have limitations:

1. Overfitting: Decision trees can readily learn the noise in the training data and generalize poorly on fresh data.
2. Unstable: Decision trees are unstable, so slight data changes can produce alternative decision trees and forecasts.

3. Bias toward particular features: Decision trees can be biased toward features higher in the tree, which can lead to inferior forecasts.

Pruning, ensemble approaches, and random forests can overcome these constraints.

2.1.3 Random Forests

Random Forest is a popular machine learning method used for both classification and regression problems. It belongs to the family of ensemble learning methods, which mix various models to improve their predictive power. In the case of Random Forest, the algorithm builds a large number of decision trees, each learned from a subset of the available data and using a random subset of the available features. The final prediction is then made by combining the guesses of all the trees in the forest [14].

One of the main benefits of Random Forest is that it is able to handle high-dimensional data with many features and can identify interactions between them. It is also less prone to overfitting than single decision trees, as the chance in the selection of data and features ensures that each tree is different and less likely to remember the training data.
Random forests have been widely used in a range of applications, such as picture classification, text analysis, and bioinformatics. However, they can be computationally expensive and may require careful setting of the hyperparameters to achieve maximum performance.

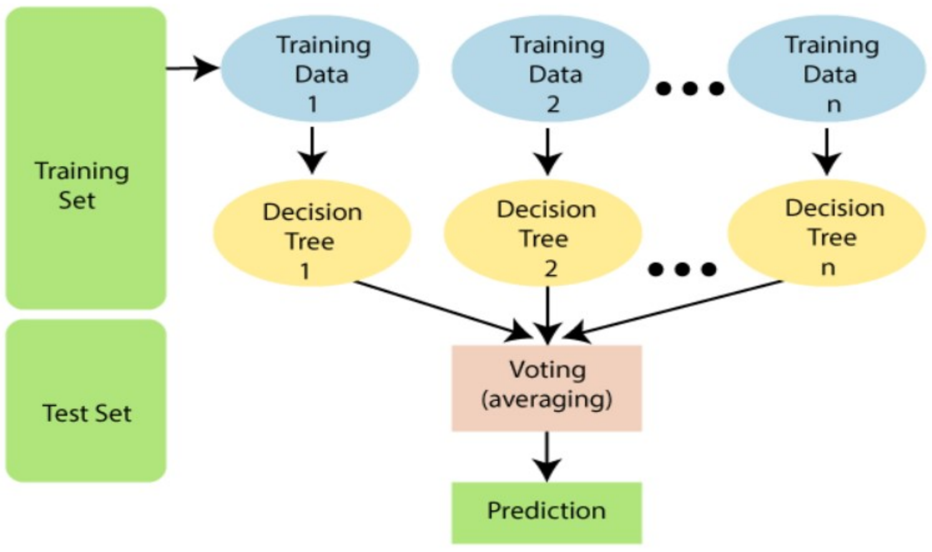


Figure 2.8: Random Forest Trees

2.1.4 Support Vector Machine(SVM)

Support Vector Machines (SVM) is an algorithm for machine learning that is used for jobs like classifying and predicting. The algorithm works by finding a hyperplane in a place with a lot of dimensions that divides the data points into different classes in the best way[15].

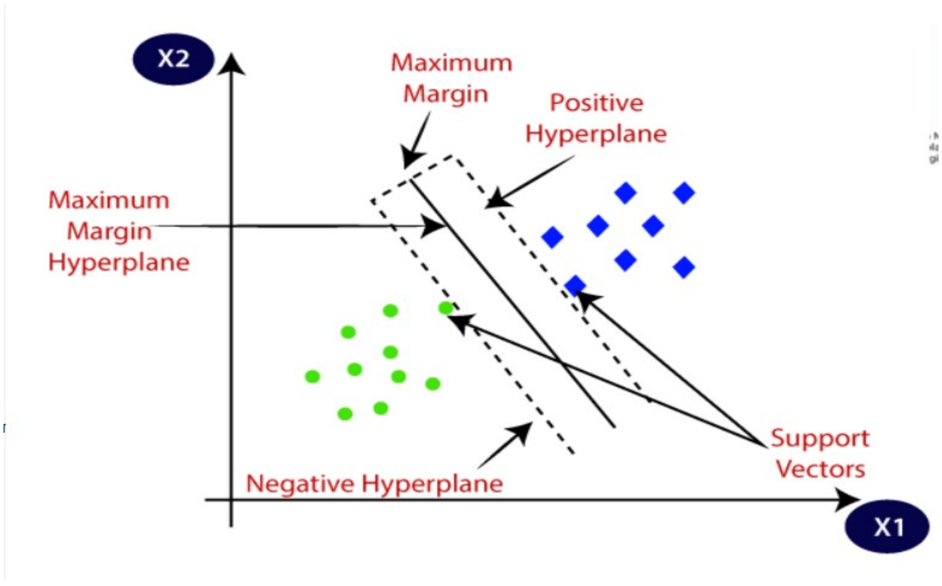


Figure 2.9: SVC Diagram

The main reasons why you should use SVM are:

1. Effective in high-dimensional spaces : SVM works well where other algorithms might fail in high-dimensional spaces.

2. When the number of dimensions is greater than the number of samples, SVM works well. This makes it useful for working with big datasets.

3. Kernel trick : SVM uses a kernel trick to move the data into a region with more dimensions, which can make the algorithm more accurate.

4. SVM is flexible because it can be used both for classification and regression.

5. Robust to outliers : SVM is less affected by outliers than other methods, like logistic regression.

But SVM has some problems as well:

1. Heavy on the computer : SVM can be hard on the computer, especially when working with big datasets.

2. Sensitivity to the choice of kernel function: The success of SVM depends on the choice of kernel function, and it can be hard to pick the right one.

3. Hard to understand : SVM gives you a binary choice boundary, which can be hard to understand.

4. Overfitting : It's easy for SVM to fit the training data too well, which means it learns the noise in the data and does a bad job of generalizing to new data.

To get around these problems, different methods can be used, such as regularization, cross-validation, and ensemble methods.

2.1.5 Multilayer Perceptron (MLP)

Deep learning uses Multilayer Perceptron (MLP) neural networks for classification and regression. It has input, hidden, and output neurons.
MLP is ideal for complicated issues because it can simulate non-linear input-output interactions. MLP can also generalize from huge datasets[16].

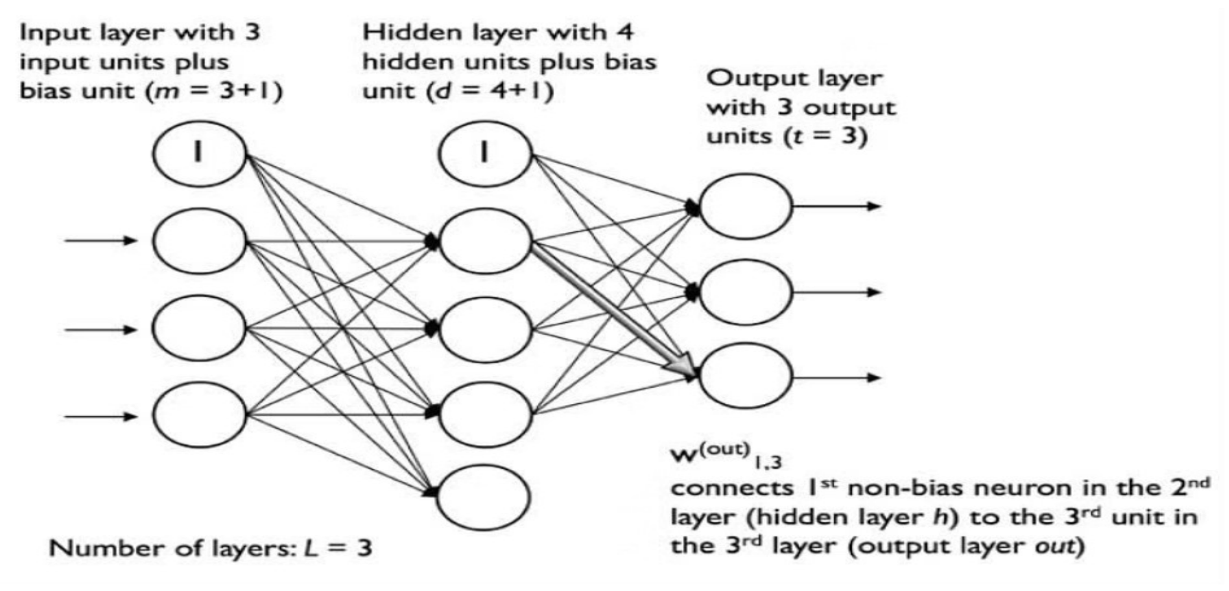

Input layer with 3 input units plus bias unit (m = 3+1)
Hidden layer with 4 hidden units plus bias unit (d = 4+1)
Output layer with 3 output units (t = 3)
1
1
$w^{(out)}_{1,3}$ connects 1st non-bias neuron in the 2nd layer (hidden layer h) to the 3rd unit in the 3rd layer (output layer out)
Number of layers: L = 3

Figure 2.10: MLP Diagram

MLP comprises:

1. MLP starts with neurons. Each neuron receives information, activates, and outputs.
2. Layers: MLP has input, hidden, and output neurons. Weights link layers.
3. An activation function adds non-linearity to neuron output. Sigmoid, ReLU, and tanh are common activation functions.
4. Backpropagation trains MLP. It reduces mistakes by propagating them back across the network and changing weights.

MLP's key challenges:

1. Overfitting: MLP easily overfits training data, learning the noise and generalizing badly on fresh data.
2. Choosing the correct architecture, including the number of layers and neurons per layer, can be difficult and affect MLP performance.
3. Choosing the correct hyperparameters, like the learning rate, regularization parameter, and batch size, can also be difficult and affect MLP performance.

Regularization, dropout, early halting, and cross-validation can address these issues.

2.1.6 XGBoost Classifier

XGBoost is a famous machine learning algorithm for solving classification, regression, and ranking problems. It stands for "Extreme Gradient Boosting" and is based on gradient boosting, a method that involves training weak classifiers on different parts of the data and combining their results to make a more accurate end model.
XGBoost is known for being fast, scalable, and accurate, and it is used a lot in business and in schools. It has won many events on sites like Kaggle and has been used in many different fields, such as finance, healthcare, and e-commerce. In the setting of classification, XGBoost builds a group of decision trees that are trained to predict the class labels of the input data. It works by minimizing a loss function that measures the difference between the predicted class labels and the real class labels. Complex models are also punished to prevent them

from being too good.

XGBoost has a number of hyperparameters, like the learning rate, the maximum depth of the trees, and the regularization strength, that can be changed to improve how well the model works. It also has early stopping, which lets the algorithm stop training immediately when the validation loss stops getting better. This prevents the model from becoming too good [17]. The XGBoost objective function for binary classification problems is given in Equation 2.4:

$$\text{obj} = \sum_{i=1}^{n} w_i \log(1 + \exp(-y_i \hat{y}_i)) + \lambda \Omega(f) \tag{2.4}$$

where:

n is the number of instances in the dataset

w_i is the weight assigned to the ith instance

y_i is the true label of the ith instance, which takes values of either -1 or 1

$\hat{y}_i$ is the predicted label of the ith instance using the current model, which is a real number between $-\infty$ and $+\infty$ λ is the regularization parameter that controls the strength of the regularization

$\Omega(f)$ is the regularization term that penalizes the complexity of the model f

2.2 Performance metrics for Classification

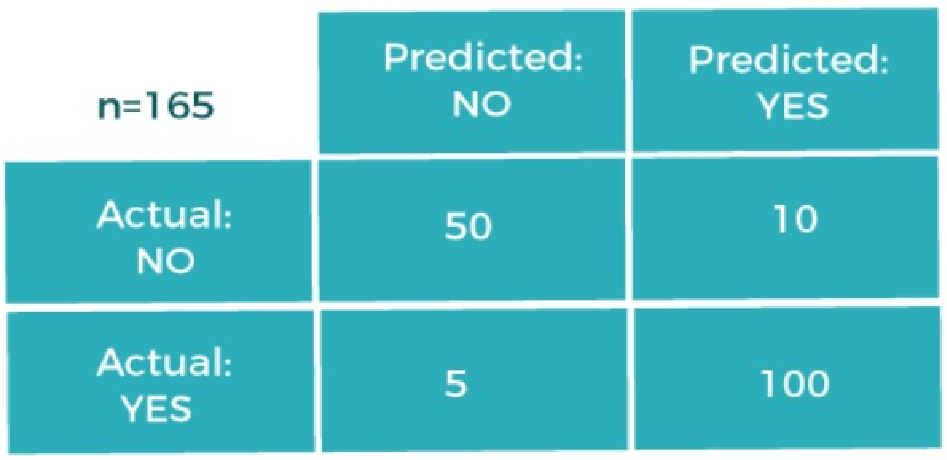
Performance review is an important part of machine learning because it lets us figure out how well a model is doing. The main goal of performance evaluation is to guess how well the model will work with new data that hasn't been seen before. There are a number of ways to measure how well a machine learning model works, such as:

1. **Accuracy**:In machine learning, accuracy is often used as a way to measure how well a classification model works. It shows what percentage of the test set cases the model correctly classifies as shown in Equation 2.5.

   Here's how to figure out how accurate a model is: :

$$Accuracy = \frac{Number of correct Predictions}{Total Number of Predictions} \tag{2.5}$$

2. **Confusion Matrix**:The effectiveness of a machine learning model for a given classification issue can be measured with the use of a confusion matrix. True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are summed up in a 2x2 matrix given a set of model predictions as shown in Figure 2.11.



n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

Figure 2.11: Confusion Matrix

- **True Positive(TP)**: The number of cases that are really positive and that the model correctly labels as positive.
- **True Negative(TN)**: The number of cases that are negative and that the model correctly identifies as negative.
- **False Positive(FP)**: The number of times the model gets it wrong and says something is positive when it should be negative.
- **False Negative(FN)**:The number of instances that are actually positive but are incorrectly classified as negative by the model.

3. **Precision**:Precision is a measure of how well a binary classification model works that is used in machine learning. It counts the number of true positives (TP) out of all the cases that the model said were positive as shown in Equation 2.6.
Here's how to figure out the precision of a model:

$$Precision = \frac{TP}{TP + FP} \tag{2.6}$$

4. **Recall**:In machine learning, recall is a performance measure that is used to judge how good a binary classification model is. It measures how many true positives (TP) there are out of all the good situations as shown in Equation 2.7.
   Here's how to figure out the recall of a model:

$$Recall = \frac{TP}{TP + FN} \tag{2.7}$$

5. **F1-Score**:In machine learning, F1-score is a performance measure that is used to judge how good a binary classification model is. It gives a single number that balances the trade-off between precision and recall. It is the harmonic mean of precision and recall as shown in Equation 2.8.
   Here's how to figure out the f1-score of a model:

$$F1 - Score = 2 * \frac{Precision * Recall}{(Precision + Recall)} \tag{2.8}$$

6. **AUC(Area Under the Curve)-ROC**:Area Under the Receiver Operating Characteristic Curve (UC-ROC) is a performance variable used in machine learning to measure how good a binary classification model is. It measures how well the model can tell the difference between positive and negative classes at different levels of chance.
   The ROC curve plots the true positive rate (sensitivity) against the false positive rate (1-specificity) for different probability thresholds. The AUC-ROC is the area under this curve, which ranges from 0.5 (random classification) to 1 (perfect classification).
   True Positive Rate(TPR) is recall only and it is calculated as 2.9:

$$TPR = 2 * \frac{TP}{(TP + FN)} \tag{2.9}$$

   False Positive Rate(FPR) can be calculated by formula 2.10:

$$FPR = 2 * \frac{FP}{(FP + TN)} \tag{2.10}$$

   If the area under the receiver operating characteristic curve (AUC-ROC) is large, it means that the model successfully classified the vast majority of positive examples with a low proportion of false positives. A balanced evaluation of the model's performance is provided, which is especially helpful when the dataset is imbalanced, i.e. when the number of cases in each class is not equal as shown in Figure 2.12.

Applications where the cost of false positives and false negatives may vary greatly include medical diagnosis, fraud detection, and credit scoring.

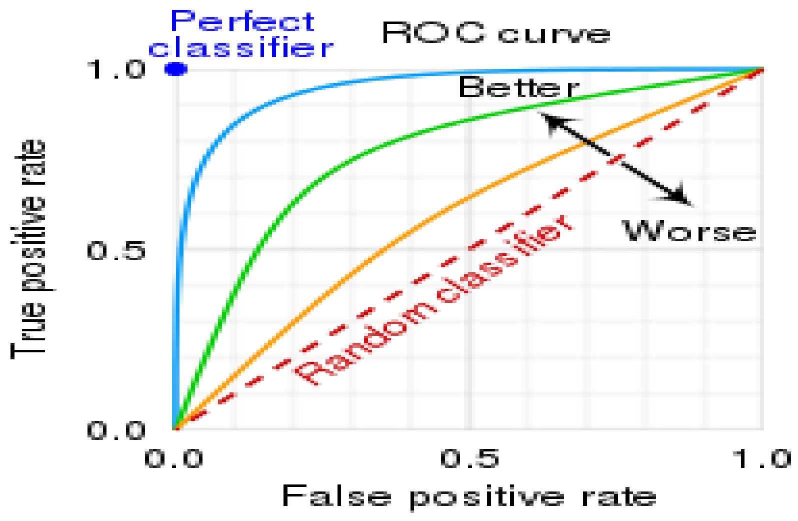


Figure 2.12: AUC-ROC Curve

In this chapter, we have described related works about phishing detection, different learning algorithms for phishing detection, and performance metrics in classification. In the next chapter, we will be describing the proposed methodology and its various subtopics, like URL collection, feature extraction, and dataset creation.

Chapter 3

Proposed Methodology

This chapter refers to our plan or approach for doing the study. It shows the steps, processes, and methods that will be used to collect and analyze data and fulfill our objectives. The flow of the phishing detection is given in Figure 3.1

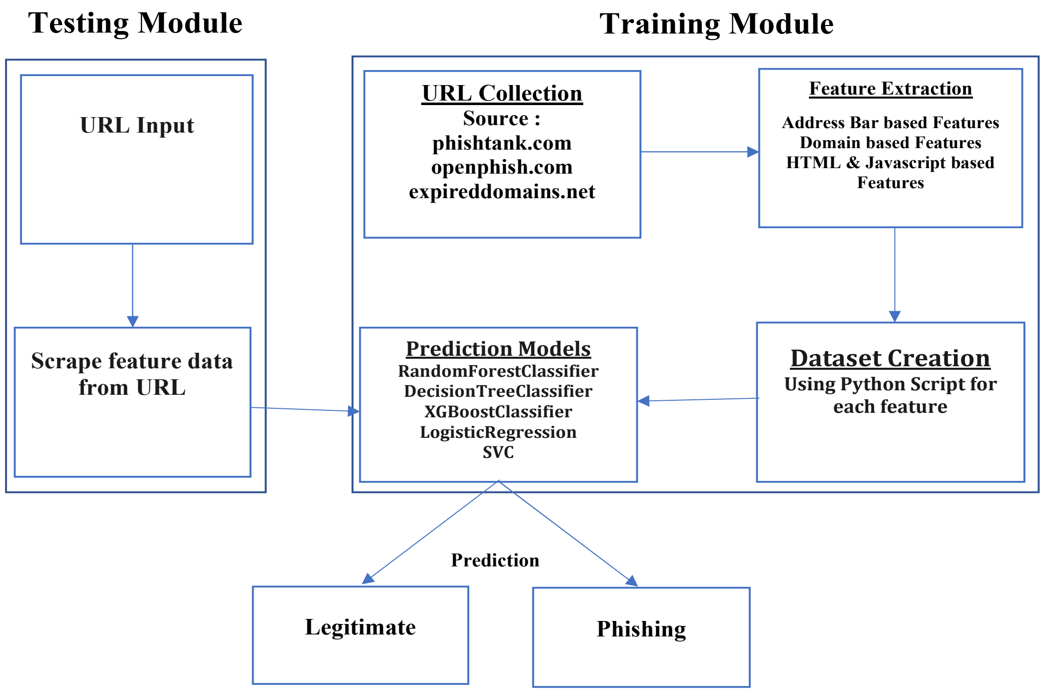


Figure 3.1: Flowchart of phishing detection

The following topics, like URL collection, feature extraction, and dataset creation, will be covered: discussed below in detail.

3.1 URL Collection

A total of 50,917 phishing websites from phishtank.com[18] and 499 phishing websites from openphish.com[19] were collected. This was possible because phishing websites have a tendency to be taken down by the phishers in a very short period of time. Because phishing websites have such a short lifespan, they are unable to be listed as top websites on repositories

such as majestic.com and alexa.com or as top websites at all. Therefore, we collected a total of 8,19,599 websites from expireddomains.net and believed that they were real [20]. This allowed us to compile a list of websites that are considered trustworthy.
During the phase of the process where the data is being collected, all of the links are initially collected in a csv file. All these sources offer filename.csv files to respectable websites as a service to the online community. PhishTank.com and OpenPhish.com, on the other hand, offered a list that had just a certain number of URLs for phishing websites each day. As soon as a sufficient number of genuine and phishing URLs were collected, the process of feature extraction by utilizing the python scripts to crawl the URLs began, and the features were extracted that are described in the next section.

3.2 Features Extraction

After going through past research papers and some of the collected websites were opened and it's html content were analysed manually. A total of 34 features were included in this study. A total of 32 out of 34 were taken from past research papers. Two features like subpagescount and totalindexedpages were added by us. These features are based on URL of the websites examined, it's domain records and HTML code. URL obfuscation is one of the many ways that attackers use to deceive the users. For example an attacker might use faceboook.com to spoof the facebook website hosted at facebook.com[11]. Mohammad et al. [10] found out 30 effective features for phishing detection in their research. In this work, we made our own dataset based on their descriptions.
The collected features are grouped into the following groups according to [10]:

- Address Bar based Features
- Domain based Features
- HTML and Javascript based Features

3.2.1 Address Bar Based Features

1. **Have_IP**: URLs that just contain an IP address, such as `http://192.88.30.124/login.php`, are almost certainly malicious attempts to steal users' credentials. The IP address is sometimes converted to hexadecimal form.

Rule: So this is a binary feature. If the domain part has an IP address, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

NOTE: This feature was not implemented during training because it is not applicable to FHD URLs. It's only for study purposes.

2. **Have_HTTPS**: The inclusion of HTTPS in a URL indicates that the connection is encrypted and secure. The vast majority of trusted sites in our data set loaded using the secure HTTPS protocol and an SSL certificate.

   Rule: So this is a binary feature. If the domain part has an HTTP scheme, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

3. **Have_At**: The "@" character in a URL causes the browser to disregard everything before it, and the genuine address is usually found after the "@" symbol.

   Rule: So this is a binary feature. If the domain part has the "@" symbol, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

4. **URL_Length**: Phishers can mask the suspicious portion of a URL by making it too long.

   Rule: So this is a continuous feature. The value of this feature will be the length of the URL.

5. **URL_Depth**: This feature indicates how far the given URL is from its home page. The depth is calculated by the "/" present in the URL.

   Rule: So this is a continuous feature. The value of this feature will be the number of "/" present in the URL.

6. **Redirection**: To indicate that the user will be taken to a different website, "//" must be included in the URL path. To illustrate a URL, consider this pair: `https://leetcode.com//https://codeforces.com/`. We look at the place where

the ”//” occurs. If the URL prefix is ”HTTP,” then the ”//” should come after the fifth dot. If ”HTTPS” is used instead, the ”//” should come seventh in the URL.

Rule: So this is a continuous feature. The value of this feature will be the number of “/” present in the URL.

7. **http_https_Domain**: In some cases, phishers will alter the domain portion of a URL by appending the ”HTTPS” token. Try this: `http://https-practice.geeksforgeeks.org/explore`

   Rule: So this is a binary feature. If the domain part has the ”HTTPS” symbol, it’s phishing, and the feature value will be 1. If not, it’s real, and the feature value is 0.

8. **TinyURL**: URL shortening is a technique used on the ”World Wide Web” that allows lengthy URLs to be shortened significantly while still directing users to the intended web page. This is achieved by setting up a ”HTTP Redirect” on a shorter domain name, which then points to the longer URL of the destination webpage. For instance, ”bit.ly/85HUS44j” is short for the URL `https://www.codechef.com/`.

   Rule: So this is a binary feature. If the shortening service has been applied, it’s phishing, and the feature value will be 1. If not, it’s real, and the feature value is 0.

   NOTE: This feature was not implemented during training because it is not applicable to FHD URLs. It’s only for study purposes.

9. **Prefix_Suffix**: The usage of hyphens (-) in web addresses is frowned upon. To trick consumers into thinking they are dealing with a legitimate website, phishers often append prefixes or suffixes to the domain name separated by (-). For instance, you may go to `http://www.securedomain-cric-buzz.com/`.

   Rule: So this is a binary feature. If domain parts contain the ”-” symbol, it’s phishing, and the feature value will be 1. If not, it’s real, and the feature value is 0.

10. **SubDomainCount**: Let’s use the link that was given as an example: `https://`

`www.mnit.ac.in/students/course`. Country-code In a domain name, country code top-level domains (ccTLDs) like ”in” can be used. The website name is ”mnit.ac,” which stands for ”mnit” and ”academic.” ”ac.in” is an SLD, which stands for ”second-level domain.” Before we can start making the rule that will pull this information out, we need to take the subdomain (www.) out of the URL. If there is a ccTLD, we need to get rid of it. The last step is to add up all the dots. If there is more than one dot, this is called a ”suspicious” URL.

Rule: So this is a binary feature. If domain parts contain dot symbol greater than 1, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

3.2.2 Domain Based Features

1. **DNS_Record**: The WHOIS database can tell if a website is a phishing site in one of two ways: either the claimed identity is not recognized by the database, or the hostname has no records.

   Rule: So this is a binary feature. If the DNS record is missing or cannot be discovered, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

2. **AgeOfDomain**: It's the point in time after a domain's creation but before its expiration.The WHOIS registry can be queried for this information. Most phishing sites don't last long before they disappear. Our analysis shows that a domain must be at least 6 months old to be considered legitimate.

   Rule: So this is a binary feature. If the age of the domain is less than 6, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

3. **Domain_End**: The interval between the domain's expiration date and the present time.The WHOIS database provides a means of obtaining this information. Most phishing sites don't last long before they disappear. Our data analysis shows that the

youngest age a domain may be to still be considered valid is now 6 months.

Rule: So this is a binary feature. If the age of the domain is less than 6, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

3.2.3 HTML and JavaScript based Features

1. **iFrame**: HTML's IFrame tag allows an external website to be shown within the current page's framework. The "iframe" tag can be used by phishers to hide a frame by hiding its boundaries and rendering it invisible. Here, the "frameBorder" element is used by the phishers. If this attribute is present, the browser will create a border around the page's content.

   Rule: So this is a binary feature. If an iframe has been used, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

2. **Mouse_Over**: It is possible for phishers to use JavaScript to display a bogus URL in the address bar. This function can be retrieved by inspecting the "onMouseOver" event in the page's source code to see if it modifies the status bar.

   Rule: So this is a binary feature. If On-Mouse-Over changes the status bar, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

3. **Right_Click**: With the help of JavaScript, phishers can prevent users from seeing or saving a website's source code by disabling the right-click menu. The functionality of this feature is same as "Using onMouseOver to hide the link." Nevertheless, right-click disabling can be checked for by checking for "event.button==2" in the page's HTML.

   Rule: So this is a binary feature. If the right-click button is disabled, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

4. **Web_Forwards**: The frequency with which a website changes URLs can be an indicator of whether or not it is a phishing site. We found that genuine websites in our dataset were redirected no more than once. However, at least four different websites

have fallen victim to this phishing technique.

Rule: So this is a binary feature. If a website is forwarded more than twice, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

5. **Google_Index**: The presence of a page in Google's index is verified by this test.Search engine results include a webpage once Google has analyzed it. Many phishing sites don't survive long enough to be indexed by Google because of this.

   Rule: So this is a binary feature. If the given URL is not indexed by Google, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

6. **Favicon**: A "favicon" is a specialized icon that represents a single website. The favicon is displayed in the address bar of many user agents, including image browsers and newsreaders, to serve as a visual reminder of the website's name. It's a red flag that the page is trying to steal personal information if the favicon doesn't match the domain in the URL bar.

   Rule: So this is a binary feature. If the favicon is loaded from an external domain, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

7. **CopyRightSymbol**: By looking for the copyright symbol, the end user can find material that can be trusted. A lot of real websites show the copyright logo on their pages to show that they own the brand for their company's name.

   Rule: So this is a binary feature. If the copyright emblem is not present, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

8. **CopyRightYear**: People often look at the copyright date to see if the site is up or not. Some websites have the copyright symbol and only the most current year, while others have a range of years from when they were first released.

   Rule: So this is a binary feature. If the copyright year is not the current year, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

9. **Domain_Around_CopyRight**: The domain name can appear either before or after the copyright indication; however, it more commonly appears after.

   Rule: So this is a binary feature. If the domain name is found around the copyright symbol, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

10. **LinkRatio**: The quantity of links to a page, even if some are from the same domain, is a good indicator of the page's trustworthiness. It counts the percentage of the page's hyperlinks that lead to the domain, often known as the "link ratio."

    Rule: So this is a ratio feature.

11. **SSL_Domain_Name**: The SSL/TSL certificate's "issued to" section displays the certificate's designated domain name. When the SSL name doesn't correspond to the domain entered in the browser's address bar, this is known as an SSL common name mismatch.

    Rule: So this is a binary feature. If the domain name mismatches with the SSL-issued name, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

12. **Submit_Handler**: Empty strings or "about:blank" in SFHs are dubious because the information provided should be used to perform some action. Additionally, if the domain name in SFHs differs from the domain name of the page, this indicates that the page is dubious because external domains rarely handle information.

    Rule: So this is a binary feature. If SFH="about:blank", it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

13. **FormMail**: A user can transmit his personal information to a computer via an online form, which then processes it. The phisher's personal email could obtain the user's information. This could be accomplished with a server-side programming language, such as PHP's "mail()" function. This could also be accomplished using the client-side

”mailto:” function.

Rule: So this is a binary feature. If ”mail()” or ”mailto:” functions to submit user information, it’s phishing, and the feature value will be 1. If not, it’s real, and the feature value is 0.

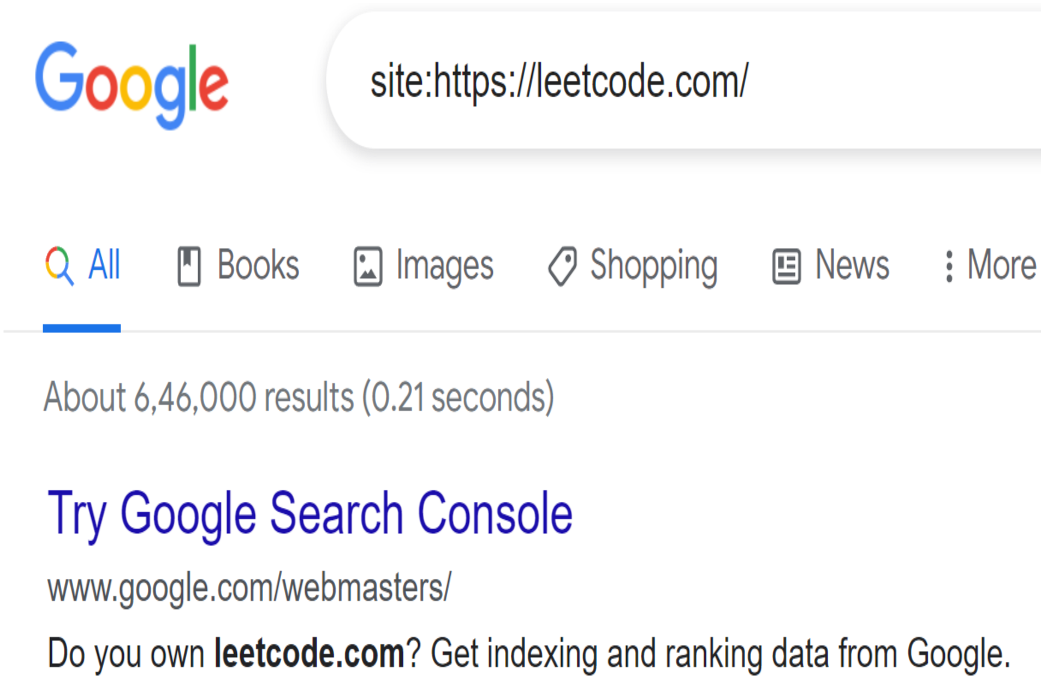
14. **NonStandardPort**: This technique can be used to check the availability of a given server’s services, such as HTTP. In order to keep out potential dangers, you should only enable the necessary ports. Some network security tools, such as firewalls, proxy servers, and network address translation servers, include settings that prevent them from opening any ports unless explicitly allowed. Users’ data is vulnerable to phishers if they can start whatever service they like. Both HTTP (port 80) and HTTPS (port 443) are required for communication.

    Rule: So this is a binary feature. If a port other than 80 or 443 is used by the given URL, it’s phishing, and the feature value will be 1. If not, it’s real, and the feature value is 0.

15. **SubPagesCount**: This feature generally calculates the number of subpages that can be visited from the home pages, and it is generally seen that phishing urls have the fewest number of subpages as compared to legitimate URLs. Examples of subpages are the About page,Contact page, Service page, etc.

    Rule: This is just a count of subpages, and it is a continuous value.

16. **IndexPagesCount**: This feature generally calculates the number of indexed pages related to the given url by the search engine as shown in Figure 3.2, and it is generally seen that phishing urls have the lowest number of total indexed pages as compared to legitimate uRLs.


site:https://leetcode.com/

All Books Images Shopping News More

About 6,46,000 results (0.21 seconds)

Try Google Search Console

www.google.com/webmasters/

Do you own **leetcode.com**? Get indexing and ranking data from Google.

Figure 3.2: Count of Indexed Pages

Rule: This is just a count of subpages, and it is a continuous value.

17. **MLSTagsRatio**: After reviewing all the possible elements for a webpage's source code, we can say that meta tags, which provide information about the HTML document, script tags, which create a client-side script, and link tags, which retrieve external web resources, are all commonly utilized by respectable websites. All of these tags should point to the same location on the page.

    Rule: This is the ratio type of value.

18. **Anchor_URL**: This features the ratio of the number of external domains loaded by the total number of URLs loaded through the anchor tag.

    Rule: This is the ratio type of value.

19. **RequestUrl**: Request URL verifies whether external objects, such as images, videos, and audio, are loaded from a different domain. The majority of items incorporated in a legitimate website share the same domain as the website itself.

Rule: This is the ratio type of value.

20. **Empty_Link**: This features the ratio of the number of external domains loaded with a void href by the total number of URLs loaded through the anchor tag.

    Rule: This is the ratio type of value.

21. **visibilityMode**: Free hosting domains typically contain a banner in the site's header or footer stating that they are free. Phishers make subtle alterations to the website's code in order to conceal the banner. For instance, the div tag, which contains the banner code, could be given the visibility:hidden attribute. This adds a veneer of legitimacy to their assaults.

    Rule: So this is a binary feature. If visibility: hidden is present in any tag hiding the name of a free domain provider, it's phishing, and the feature value will be 1. If not, it's real, and the feature value is 0.

The features have been given below for quick lookup. Address bar based features are in Table 3.1, domain based features in 3.2 and HTML and javascript based features in Table 3.3.

Table 3.1: Address Bar based Features

Sno.	Feature Name	Description	Type
1	HTTPS Scheme	Checks the presence of HTTPS Scheme in URL	Binary
2	'@' Symbol	Checks the presence of '@' in URL	Binary
3	URL Length	Checks the length of URL	Continuous
4	URL Depth	Checks how far the given URL is from it's home page	Continuous
5	Redirection	Checks the presence of '//' in URL after 7th position	Binary
6	HTTPS present in Domain	Checks the presence of 'HTTPS' in domain of URL	Binary
7	Prefix Suffix by '-'	Checks the presence of '-' in domain of URL	Binary
8	Sub Domain Count	Checks the number of '.' in domain of URL	Binary

Table 3.2: Domain based Features

Sno.	Feature Name	Description	Type
1	DNS Record	Checks the identity of URL using whois database	Binary
2	Age of Domain	Checks the total age of domain in months	Binary
3	Domain End	Checks the total age left of domain in months	Binary

Table 3.3: HTML and Javascript based Features

Sno.	Feature Name	Description	Type
1	Iframe	Checks the presence of any phishing websites embedded using iframe tag in given URL	Binary
2	Mouse Over	Checks the the hovering of phishing URLs in websites	Binary
3	Right Click	Checks whether rightclick is on or off in websites	Binary
4	Web Forwarding	Checks the number of times the given URL is redirecting	Binary
5	Google Index	Checks whether the given URL is indexed by google or not	Binary
6	Favicon	Checks whether favicon is loaded internally or from external domain	Binary
7	Copyright Symbol	Checks the presence of copyright symbol in website footer content	Binary
8	Copyright year	Checks the presence of copyright year in website content to be of current year	Binary
9	Domain name around copyright	Checks the presence of domain name around copyright	Binary
10	Link Ratio	Checks how many href links in website points to itself	Ratio
11	SSL name and domain name	Checks SSL issued name and domain name to be same	Binary
12	Server Form Handler	Checks whether the form is being submitted to about:blank	Binary
13	Form mail	Checks whether the form is being submitted to mailto function	Binary
14	Non standard port	Checks whether the website uses port other than 80 and 443	Binary
15	Sub pages count	Checks the number of web pages inserted in home page of the URL	Continuous
16	Indexed pages count	Checks the number of web pages related to the URL indexed by any search engine	Continuous
17	MLT tags ratio	Checks how many links in MLT tags loaded from external domain	Ratio
18	Anchor URL	Checks how many links loaded from external domain	Ratio
19	Request URL	Checks how many links for loading images,videos and sound loaded from external domain	Ratio
20	Empty URL	Checks how many links are just empty ,not pointing to anything	Ratio
21	Visibility mode	Checks whether the banners or any advertising text of free domain providers are being hidden	Binary

3.3 Dataset Creation

Due to the fact that the study is concerned with free hosted domain (FHD)-based URLs, the URL set that has been compiled contains every conceivable kind of URL. Free hosting domain providers in Table 3.4 are mentioned by Roy et al.[1] and some more providers name of same type are added and given in Table 3.5. These domain providers are considered in this study. The necessary URLs based on the names of domain providers are filtered by matching domain providers name with the required URLs.

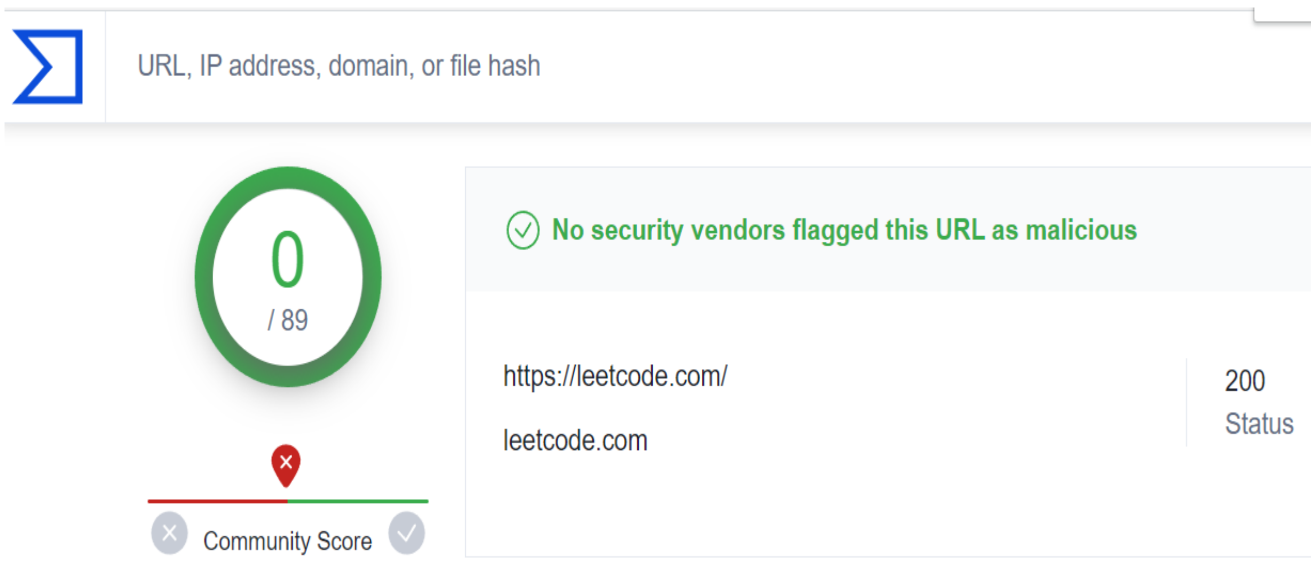
Table 3.4: Free Hosting Domains by Roy et al.[1]

Parameter / Domain Provider	PhishTank	Openphish	Alexawebsites	Total
weebly	1002	2	541	1545
duckdns	1102	2	137	1241
000webhost	195	3	2	200
blogspot	424	0	6202	6626
wix	253	0	786	1039
github	174	0	2666	2840
firebase	3324	0	290	3614
squareup	0	0	2	2
wordpress	116	0	2442	2558
sharepoint	37	0	10796	10833
yolasite	202	1	15	218
myftp	24	0	5	29
godaddysites	547	0	1	548
mailchimp	0	0	3	3
atwebpages	0	0	2	2
glitch	421	2	66	489
webnode	36	0	26	62
herokuapp	20	0	262	282
website	123	1	914	1038
netlify	175	1	411	587
hpage	0	0	22	22

Table 3.5: Some more Free Hosting Domains added during study

Parameter / Domain Provider	PhishTank	Openphish	Alexawebsites	Total
infinityfree	2	0	2	4
byethost	11	0	10	21
hyperphp	45	0	0	45
awardspace	0	0	1	1
freehostia	0	0	4	4
freehosting	0	0	2	2
freewebhostingarea	0	0	1	1
hostpapa	0	0	8	8
ultahost	0	0	1	1
porkbun	0	0	1	1
bluehost	5	0	8	13
googiehost	0	0	1	1
siteground	0	0	10	10
dreamhost	2	0	8	10
hostgator	1	0	7	8
domainracer	0	0	1	1
namecheap	0	0	3	3
inmotionhosting	4	0	1	5
a2hosting	0	0	4	4
interserver	0	0	1	1

We used the VirusTotal [21] to examine all of the URLs that we had collected, and those URLs that received a score of less than 2 were assigned the label ”legitimate URL and label 0,” while the remaining URLs were assigned the label ”phishing URL and label 1.” Virus Total is a web-based service that analyzes suspicious files and URLs with the assistance of antivirus engines and website scanners in order to determine the presence of various types of malware and other types of potentially harmful information as shown in Figure 3.3.



No security vendors flagged this URL as malicious

https://leetcode.com/

leetcode.com

200
Status

Figure 3.3: VirusTotal UI

Therefore, after applying filters, we were successful in obtaining FHD-based URLs, We were also interested in exploring the possibility of training and testing our features on general or non-FHD URLs, as well as filtering non-FHD URLs out of our collection of URLs. The count of URLs from each catagory are given in Table 3.6.

Table 3.6: Number of URL collected in each catagory

Domain Provider Type \ URL Type	Legitimate	Phishing
FHD URLs	2499	2423
Non-FHD URLs	2499	2511

In this chapter, we have described the proposed methodology and its various subtopics, like URL collection, feature extraction, and dataset creation. In the next chapter, we will be analyzing the performance of the dataset on various machine learning algorithms and comparing it with existing methodology.

Chapter 4

Performance Evaluation

In order to evaluate the efficacy of machine learning models on the most recent versions of our FHD and phishing datasets, we carried out two separate sets of experiments. On each dataset, we trained and tested six distinct machine learning models, which were covered in a previous section, and then examined the performance metrics of the machine learning model that provided the best accuracy for each dataset. These experiments are as follows:

1. Training and testing machine learning models on phishing dataset.
2. Training and testing machine learning models on FHD dataset.

4.1 Experimental Setup

In order to implement our methodology, we used the sklearn package from machine learning on Google Collab. We divided our dataset in an 80:20 ratio for train and test, respectively. Data is normalized using the Standard Scalar method from sklearn before being fed for training or testing the models. The following algorithms were implemented on the dataset for non-FHD URLs, with hyperparameter values used for each algorithm:

1. **Random Forest Classifier**: It is a popular machine learning method used for both classification and regression problems. We used n_estimators = 250, the rest parameter values were set to default, and the cost function was Gini impurity to get maximum accuracy.
2. **Decision Tree Classifier**: A prominent supervised machine learning technique for classification and regression is decision trees. We used criterion='log_loss', max_depth=6, and rest parameter values to default to get maximum accuracy.
3. **XGBoostClassifier**: Optimized distributed gradient boosting library XGBoost trains machine learning models efficiently and scalablely. We used learning_rate=0.04, max_depth=None, n_estimators=250, n_jobs=-1, and rest parameter values to default to get maximum accuracy.

4. **Multi Layer Perceptron**: Deep learning uses multilayer perceptron (MLP) neural networks for classification and regression. It has input, hidden, and output neurons. We set all parameters to default, and cost_function is a cross-entropy loss function for best accuracy.

5. **Logistic Regression**: It is a popular algorithm for machine learning that is used for jobs that can be broken down into two groups. We set all parameters to default, and cost_function is a cross-entropy loss function for best accuracy.

6. **Support Vector Machine**: SVM is an algorithm for machine learning that is used for jobs like classifying and predicting. We set all parameters to default and left the rest at default for best accuracy.

The following algorithms were implemented on the dataset for FHD URLs, with hyperparameter values used for each algorithm:

1. **Random Forest Classifier**: It is a popular machine learning method used for both classification and regression problems. We set all parameter values to default, and the cost function was Gini impurity to get maximum accuracy.

2. **Decision Tree Classifier**: A prominent supervised machine learning technique for classification and regression is decision trees. We used criterion='entropy' and rest parameter values to default to get maximum accuracy.

3. **XGBoostClassifier**: Optimized distributed gradient boosting library XGBoost trains machine learning models efficiently and scalablely. We used learning_rate=0.04, loss='exponential', max_depth=None, n_estimators=250, and rest parameter values to default to get maximum accuracy.

4. **Multi Layer Perceptron**: Deep learning uses multilayer perceptron (MLP) neural networks for classification and regression. It has input, hidden, and output neurons. We used parameters like alpha=0.001,learning_rate='adaptive',random_state=42, hidden_layer_sizes=([100,100,100])) and cost_function is cross-entropy loss function for best accuracy.

5. **Logistic Regression**: It is a popular algorithm for machine learning that is used for jobs that can be broken down into two groups. We used parameters like random_state=42, solver='saga', and cost_function is a cross-entropy loss function for best accuracy.

6. **Support Vector Machine**: SVM are an algorithm for machine learning that is used for jobs like classifying and predicting. We used the parameters kernel='rbf', random_state=42, and rest as defaults for best accuracy.

4.2 Training and testing machine learning models on phishing dataset

We performed experiments with phishing dataset on the following models and obtained accuracy, which is discussed in Table 4.1:-

Table 4.1: Performance of ML Models on phishing dataset

Model Name	Train Accuracy	Test Accuracy
Random Forest Classifier	0.999	0.977
XG Boost Classifier	0.989	0.975
Multilayer Perceptrons(MLP)	0.986	0.97
Decision Tree Classifier	0.976	0.97
Logistic Regression	0.97	0.968
Support Vector Machine	0.951	0.941

Since Random forest classifier has performed with best test accuracy so it's features importance graph is shown in Figure 4.1 , it's confusion matrix and AUC-ROC curve is shown in Figure 4.2 and performance matrics are discussed in Table 4.2.

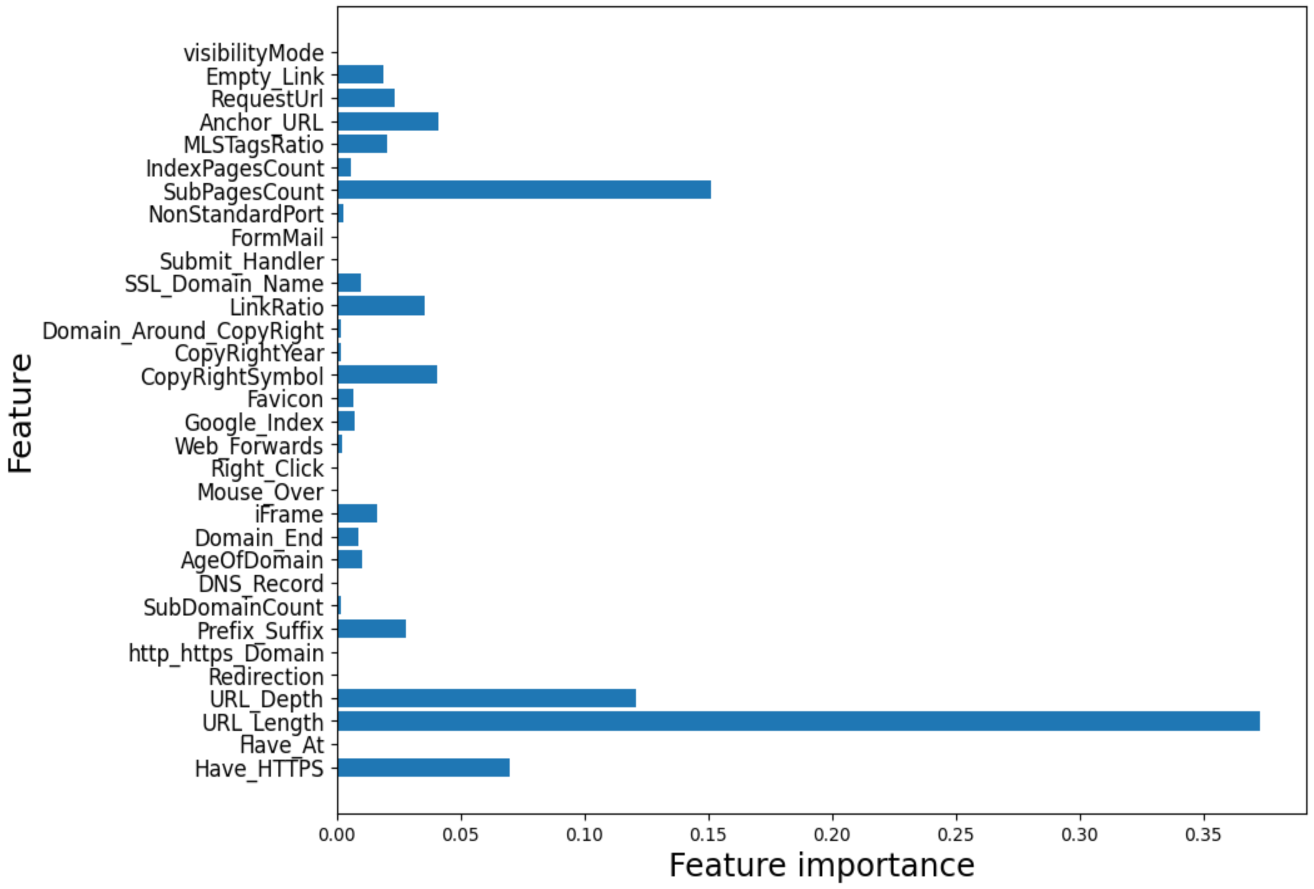
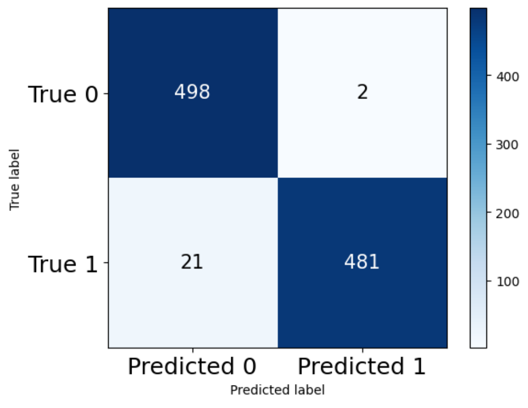
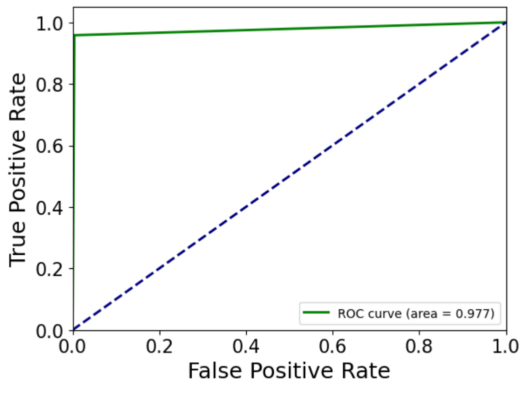


Figure 4.1: Feature Importance of phishing dataset on random forest classifier



(a) Confusion Matrix for phishing dataset



(b) AUC-ROC of phishing dataset

Figure 4.2: Confusion Matrix and AUC-ROC curve on phishing dataset

Table 4.2: Performance Metrics on phishing dataset

Model Name	Accuracy	Precision	Recall	F1-Score
Random Forest Classifier	0.977	0.996	0.958	0.977

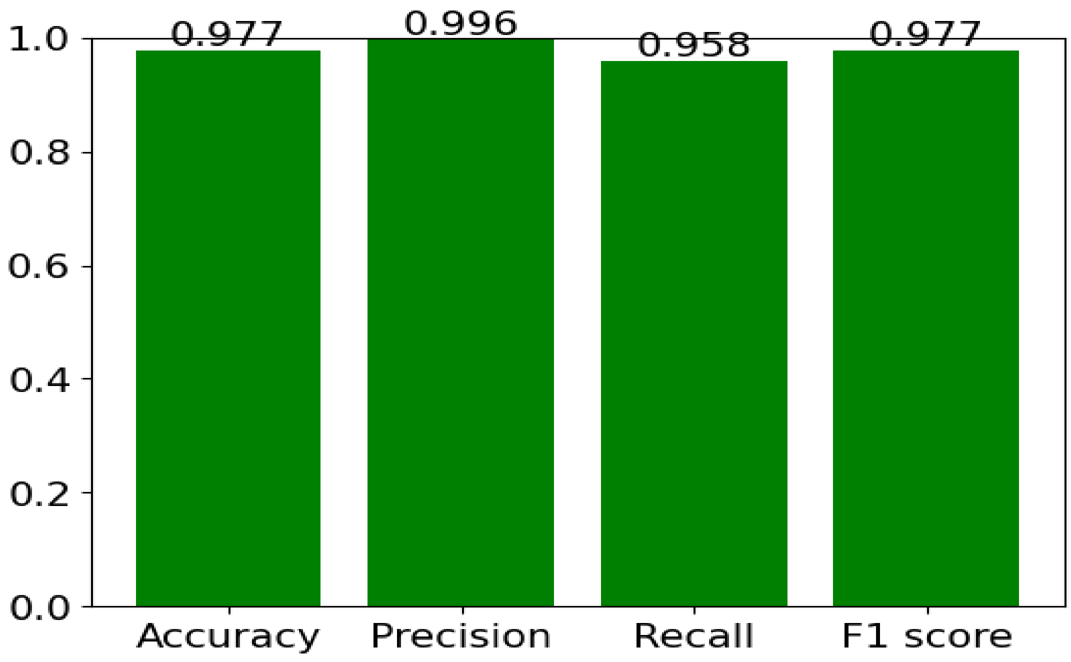


Figure 4.3: Performance metrics on phishing dataset

A detailed comparison of the proposed methodology with existing methodology Panigua et al. [1] has been done in terms of parameters like model name,number of features, size of dataset, accuracy, precision, recall, recall and f1-score in Table 4.3.

Table 4.3: Comparison of Existing Methodology with Proposed Methodology

Parameters / Methodology	Model name	No. of features	size of dataset	Accuracy	Precision	Recall	F1-score
PANIAGUA et al.2017 [8]	LightGBM Classifier	**38**	**60000**	0.947	0.953	0.939	0.946
Proposed Method	Random Forest Classifier	32	5010	**0.977**	**0.996**	**0.958**	**0.977**

4.3 Training and testing machine learning models on FHD dataset

We performed experiments with FHD dataset on the following models and obtained accuracy, which is discussed in Table 4.4:-

Table 4.4: Performance of ML Models on FHD dataset

Model Name	Train Accuracy	Test Accuracy
Multilayer Perceptrons(MLP)	0.972	0.965
Random Forest Classifier	0.982	0.955
XG Boost Classifier	0.965	0.947
Support Vector Classifier	0.932	0.937
Logistic Regression	0.919	0.93
Decision Tree Classifier	0.982	0.927

Since multilayer perceptron classifier has performed with best test accuracy so it's features importance graph is shown in Figure 4.4, it's confusion matrix and AUC-ROC curve is shown in Figure 4.5 and performance matrics are discussed in Table 4.5.

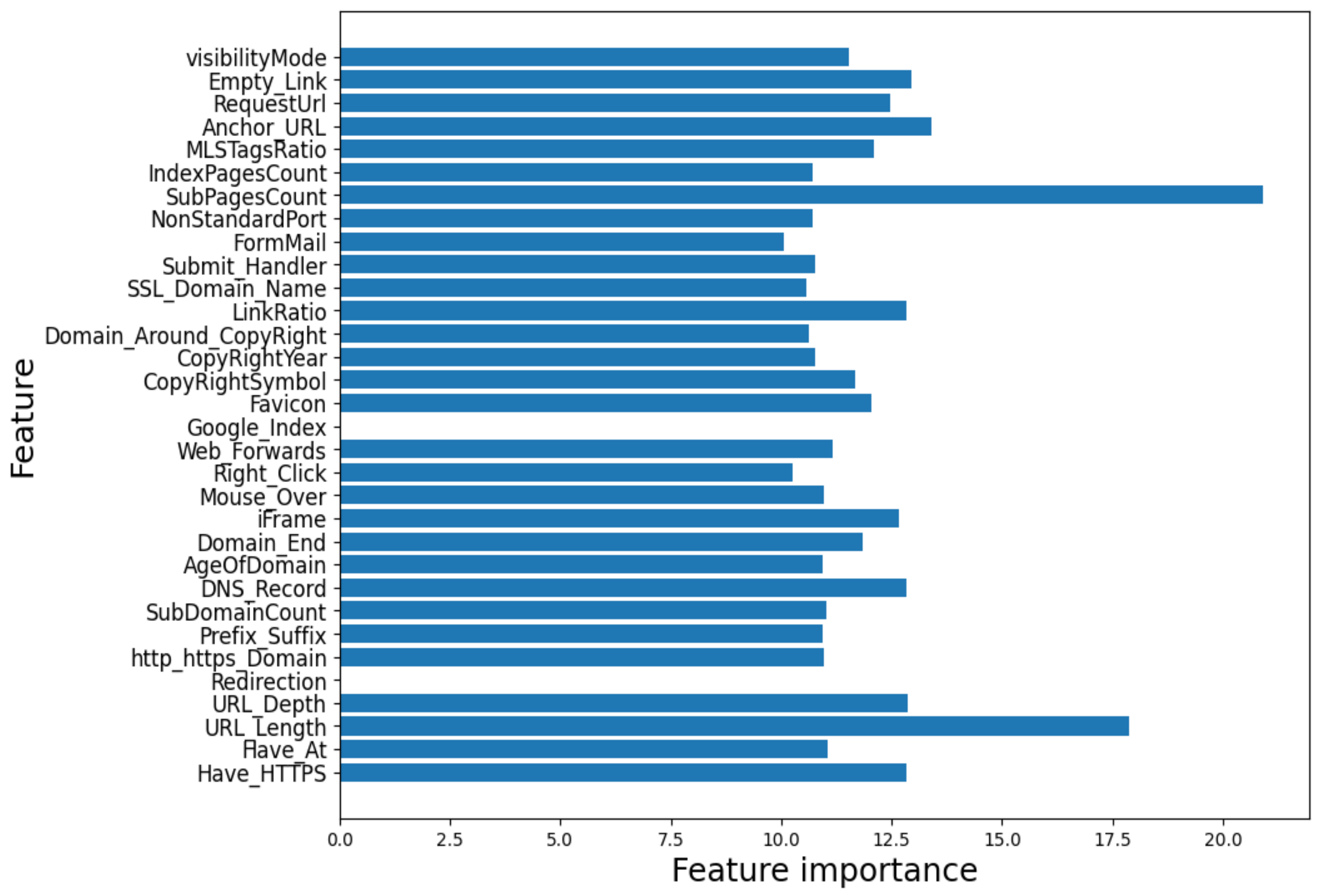
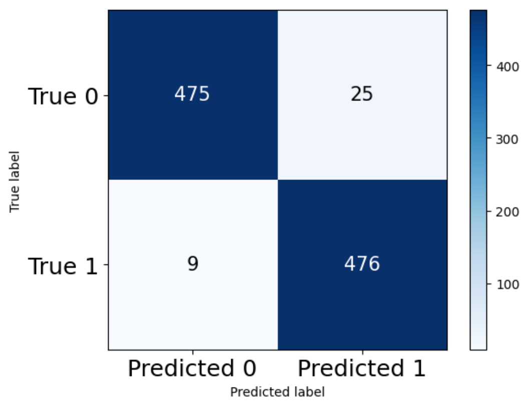
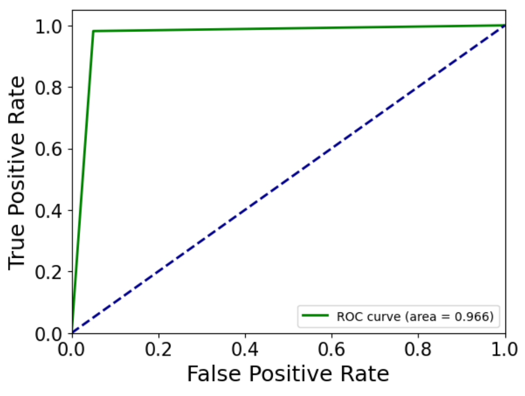


Figure 4.4: Feature Importance of FHD dataset on multilayer perceptron classifier



(a) Confusion Matrix for FHD dataset



(b) AUC-ROC of FHD dataset

Figure 4.5: Confusion Matrix and AUC-ROC curve on FHD dataset

Table 4.5: Performance Metrics on FHD dataset

Model Name	Accuracy	Precision	Recall	F1-Score
Multilayer Perceptrons(MLP)	0.965	0.950	0.981	0.966

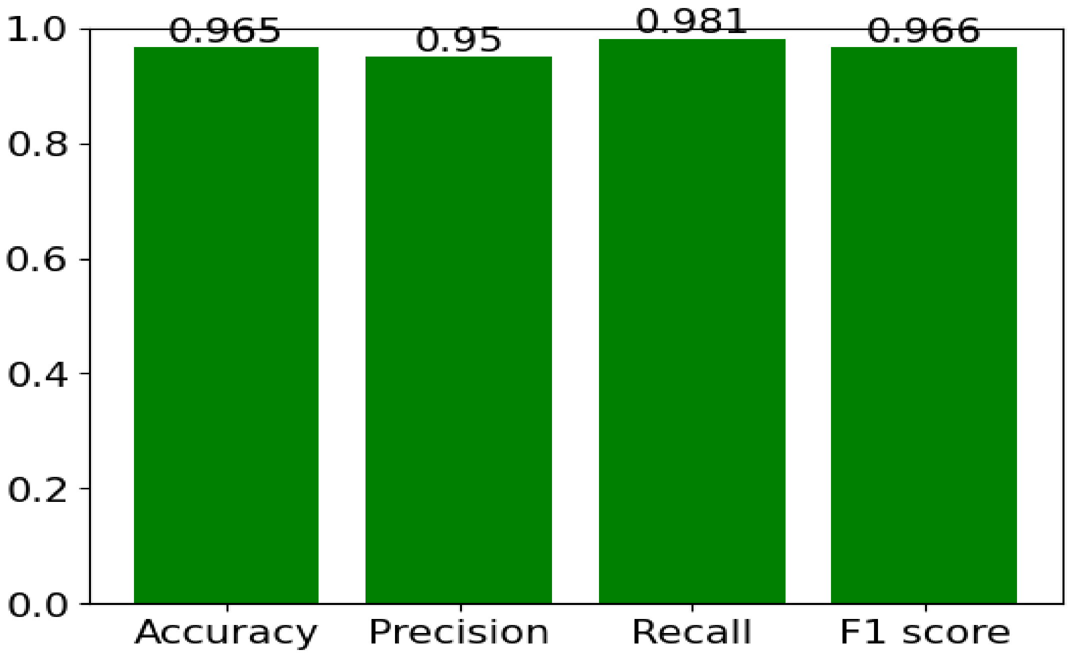


Figure 4.6: Performance metrics on FHD dataset

A detailed comparison of the proposed methodology with existing methodology Roy et al. [1] has been done in terms of parameters like model name, number of features, size of dataset, accuracy, precision, recall, recall and f1-score in Table 4.6.

Table 4.6: Comparison of Existing Methodology with Proposed Methodology

Parameters / Methodology	Model name	No.ofFHD providers	No. of features	size of dataset	Accuracy	Precision	Recall	F1-score
Roy et al.2022[1]	Random Forest Classifier	24	7	2764	**0.973**	**0.975**	0.975	0.975
Proposed Method	Multilayer Perceptron	**41**	**32**	**4922**	0.965	0.950	**0.981**	**0.996**

In this chapter, we have analyzed the performance of the datasets on various machine learning algorithms and compared it with existing methodology. In the next chapter, we will talk about our final conclusions and future works.

Chapter 5
Conclusions and Future Works

5.1 Conclusions

We identified a family of phishing websites hosted over free web hosting domains (FHDs), which can be created and maintained at scale with very little effort, while also effectively evading prevalent anti-phishing detection and resisting website takedowns. This thesis found a way to check a new kind of phishing website that is hosted on FHDs. We developed an up-to-date dataset of 4922 FHD URLs that were just recently collected. Using a multilayer perceptron neural network with 32 features, we got the best accuracy of 96.5%, covering the maximum characters of phishing websites. We also put our features to the test on a set of self-made phishing URL datasets and got a 97.7% success rate.

5.2 Future Works

Phishing detection is an ever-evolving game of cat and mouse, and the attackers always come up with more and more complex ways to deceive users. So these features might obsolete for detection of phishing attacks. New adaptive features that can be found out to detect evolving phishing attacks. New ensemble techniques could be found out that could detect phishing websites with better accuracy. Features engineering could be used to narrow down the total number of features into specific features for accurate phishing detection. Visual characteristics of phishing and legitimate websites can be extracted and utilised and convolutional neural networks cab be used to detect phishing websites using visual features. A new classification problem between phishing websites and FHD websites could be solved as a separate problem.

References

[1] Sayak Saha Roy, Unique Karanjit, and Shirin Nilizadeh. A large-scale analysis of phishing websites hosted on free web hosting domains. *arXiv preprint arXiv:2212.02563*, 2022.

[2] Top most intriguing recent phishing attacks statistics. https://www.getastra.com/blog/security-audit/phishing-attack-statistics/#:~:text=Phishing%20email%20statistics%20suggest%20that,attack%20occurring%20every%2011%20seconds. Accessed: 2023-04-24.

[3] Ibm's 2022 cost of data breach statistics. https://www.egress.com/blog/phishing/phishing-statistics-round-up. Accessed: 2023-04-24.

[4] Verizon 2022 data report. https://www.verizon.com/business/en-gb/resources/reports/dbir/. Accessed: 2023-04-24.

[5] Unifying the global response to cybercrime report 2022. https://www.comparitech.com/blog/vpn-privacy/phishing-statistics-facts/. Accessed: 2023-04-24.

[6] Insufficient phishing websites data. https://zvelo.com/single-use-phishing-urls-need-zero-second-detection/. Accessed: 2023-04-26.

[7] Hossein Shirazi, Kyle Haefner, and Indrakshi Ray. Fresh-phish: A framework for auto-detection of phishing websites. In *2017 IEEE international conference on information reuse and integration (IRI)*, pages 137–143. IEEE, 2017.

[8] Manuel Sánchez-Paniagua, Eduardo Fidalgo Fernández, Enrique Alegre, Wesam Al-Nabki, and Victor Gonzalez-Castro. Phishing url detection: A real-case scenario through login urls. *IEEE Access*, 10:42949–42960, 2022.

[9] Mustafa Kautan and Davut Hanbay. Effective classification of phishing web pages based on new rules by using extreme learning machines. *Computer Science*, 2(1):15–36, 2017.

[10] Rami M Mohammad, Fadi Thabtah, and Lee McCluskey. Phishing websites features. *School of Computing and Engineering, University of Huddersfield*, 2015.

[11] Shalin Kumar Deval, Meenakshi Tripathi, Bruhadeshwar Bezawada, and Indrakshi Ray. “x-phish: Days of future past”: Adaptive & privacy preserving phishing detection. In *2021 IEEE Conference on Communications and Network Security (CNS)*, pages 227–235. IEEE, 2021.

[12] Logistic regression theory concept. `https://www.javatpoint.com/logistic-regression-in-machine-learning`. Accessed: 2023-04-26.

[13] Decision tree theory concept. `https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm`. Accessed: 2023-04-26.

[14] Source of random forest tutorials. `https://towardsdatascience.com/random-forests-algorithm-explained-with-a-real-life-\example-and-some-python-code-affbfa5a942c`. Accessed: 2023-04-26.

[15] Support vector machine theory concept. `https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm`. Accessed: 2023-04-26.

[16] Multilayer perceptron theory concept. `https://medium.com/codex/introduction-to-how-an-multilayer-perceptron-works-\but-without-complicated-math-a423979897ac`. Accessed: 2023-04-26.

[17] Source of xgboost tutorials. `https://towardsdatascience.com/a-beginners-guide-to-xgboost-87f5d4c30ed7`. Accessed: 2023-04-26.

[18] Url collections from phishtank. `http://data.phishtank.com/data/onlinevalid.csv`. Accessed: 2023-04-05.

[19] Url collections from openphish. https://openphish.com/feed.txt. Accessed: 2023-04-05.

[20] Url collections from expireddomains. https://www.expireddomains.net/alexa-top-websites/. Accessed: 2023-04-05.

[21] Virustotal. https://www.virustotal.com/gui/home/url. Accessed: 2023-04-26.