International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

PHISCAN: Phishing Detector Plugin using Machine Learning

Sachin Barahate[1], Prachit Raut[2], Harshal Vengurlekar[3], Rishikesh Shete[4]

[1]Professor, [2,3,4] Student, Department of Computer Engineering, Vasantdada Patil Pratishthan's College of Engineering and Visual Arts, Mumbai, Maharashtra, India

---

Abstract - *Phishing URL may be a widely used and customary technique for cybersecurity attacks. Phishing is a cybercrime that tries to trick the targeted users to expose their private and sensitive information to the attacker. The motive of the attacker is to gain access to personal information like usernames, login credentials, passwords, financial account details, social networking data, and personal addresses. These private credentials are then often used for malicious activities like fraud, notoriety, gain, reputation damage, and lots of more illegal activities. This paper presents a comprehensive study of various existing systems used for phishing website detection. The system presented here uses advanced machine learning and to realize better precision and better accuracy while categorizing websites as phishing or benign.*

Key Words: Phishing, Phishing URL, Detection, Machine Learning.

1. INTRODUCTION

A successful phishing attack is a fraudulent attempt to obtain personal sensitive information. Detecting a phishing website can be difficult as the attacker imitates the original website's overall look to trick the user. This paper proposes a phishing detector plugin for fast and accurate detection of phishing websites to safeguard the user's personal and sensitive information. The plugin is developed for Chrome browser and uses technologies of JavaScript and HTML for classification of URLs at the client side. The use of JavaScript in the prediction model ensured user privacy while enabling fast and swift classification and detection time.

1.1 Phishing

Phishing is that the fraudulent plan to obtain sensitive information or data, such as usernames, passwords, and MasterCard details, or other sensitive details, by impersonating oneself as a trustworthy entity in digital communication. Typically administered by email spoofing instant messaging and text messaging, phishing often directs users to enter personal information at a fake website that matches the design and feel of the legitimate site. Phishing is an example of social engineering techniques wont to deceive users. Users are lured by communications purporting to be from trusted parties such as social networking websites, auction sites, banks, emails/messages from friends or colleagues/executives, online payment systems, or IT administrators.

1.2 Website URL

URL is the shortened form of Uniform Resource Locator, which is also known as a web address of the machine and other resources on the World Wide Web.

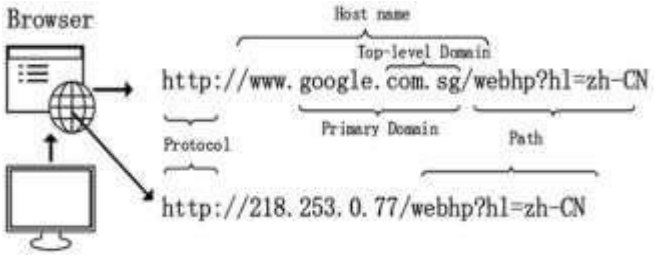


Fig -1: Website URL (Uniform Resource Locator)

As shown in Figure, URL has two fundamental components:

- To shows what convention to utilize convention identifier is used
- Resource title indicates the IP address or the space name or the trail where the asset is found. The two forward slashes and a colon isolate the resource title and the scheme name (also called as convention identifier). Compromised URLs that are utilized for cyber-attacks are termed malicious URLs while others are known as Benign URLs. In reality, it was noted that near to one-third of all websites are possibly malicious in nature, demonstrating uncontrolled utilize of malicious URLs to perpetrate cyber-crimes.

A large part of most of today's cyber-security threats is Malicious URLs. In nature, nearly one-third of all websites are potentially malicious, demonstrating the use of malicious URLs to commit cyber-crimes is widespread.

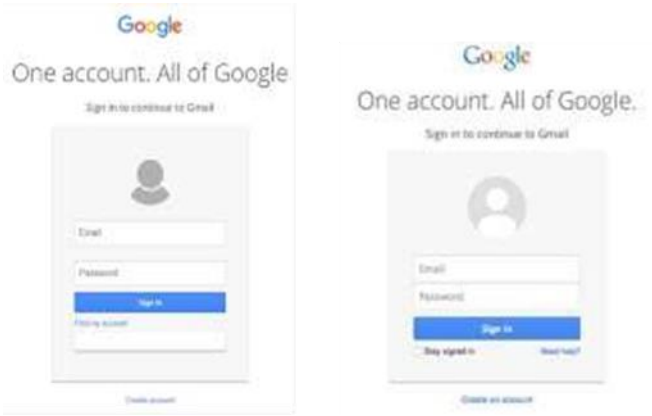
Cybercriminals also used this kind of tools:

- start phishing campaigns aimed toward stealing your personal information,
- Getting you by downloading a file, viruses or Trojans, install malware or as simple as drive-by-download prompted by something as simple as a mouse-over
- Start a spam campaign that may involve malicious advertising, scams, phishing, or other cyber fraud.

The unsolicited content hosted by these malicious website lure the target victims to fall prey to such malicious campaigns causing billions of dollars in losses every year.

1.3 Phishing Website Detection

Phishing is an attack where a legitimate user is deceived to disclose sensitive information and assets with value. Loss of such sensitive information might cause potential economic or reputational harm to an organization. Phishing uses social engineering techniques to trick users such as creating fake websites that clone with the same attributes and design as the existing legitimate one. In a classic phishing attack, a phisher sends a link enclosed in a message to the user. The link redirects the user to the cloned malicious page which looks similar to the original webpage but is not and is intended to steal the user's sensitive data. Such phishing attacks have proven to cause tons of monetary loss to varied organizations.



(a) (b)

Fig -2: An example of a Phishing attack (a) that mimics the original Gmail login (b)

Thus, phishing attacks can be prevented by exterminating such harmful websites with the help of a "Phishing Detection" tool. Machine learning is one of the powerful techniques which may make the detection of phishing websites tons simpler. A detection tool will easily classify legitimate and phishing URLs with the help of advanced machine learning algorithms.

URLs of the varied websites are separated into 3 important classes:

- Benign: These are Safe websites that provide normal services to people.
- Spam: These Websites performs flooding the user with advertising or sites such as fake surveys and online dating etc.
- Malware: These websites which are created by attackers look like normal websites can make use of sensitive contents of people.

The systems reviewed in this paper range from different detection techniques and tools used by many researchers. In these researched papers it ranges from Blacklist and Heuristic features to visual and content-based features. The studies presented here use advanced machine learning to realize better precision and better accuracy while categorizing websites as phishing or benign. The purpose of conducting the project is to detect fake websites. Web pages differ with the feature set and thus, we can use it as our prime weapon to prevent phishing attacks. The basis of the proposed work is to perform advanced machine learning algorithm-based classification for various features of the website which would have a high rate of accuracy of detecting phishing websites.

2. MOTIVATION

Phishing attacks are the source of 90% of successful cyber-attacks. URL-based attacks comprised 62% of all email-based phishing attacks in 2019. In 2019, 167% increase in HTTPS URLs hosting malicious content.

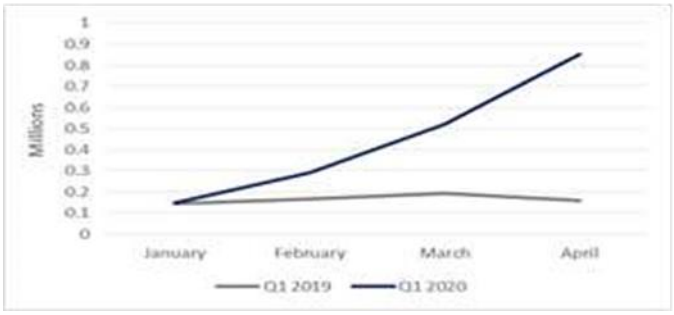


Fig -3: Phishing attacks in Q1 2019 and Q1 2020

Microsoft Windows has a dominating presence in the desktop operating system with a share of 77.74 percent and the year 2019 saw a 181% Increase in Microsoft-based phishing attacks. Also, mobile phone operating systems do not have a security mechanism or antivirus or malware detection system which makes them more prone to attacks. Currently, a lot of existing tools, encapsulated in browsers, search engines, or applications, such as SafeBrowsing from Google and SmartScreen from Microsoft, try to inform a user that a specific URL the user is about to visit has been identified as unsafe or malicious. This is realized by matching the URL being visited with blacklists constructed by the security community. Those blacklists are accumulated using various techniques, ranging from a user reporting to WebCrawler's with site content analysis to automatic classification based on heuristics or machine learning classifiers.

However, many malicious websites can still sneak through such protection systems, which can be the consequence of several reasons:

- The website is too new and thus has not been scanned or analyzed by any mechanisms yet.
- The website has been incorrectly analyzed, either due to the imperfection of mechanisms or the countermeasures against detection taken by the attackers, e.g. "cloaking", or abusing the legal short URL services.

There exists systems aiming at addressing the issue of incomplete blacklists by using real-time client-side

evaluation against the content or behavior of the website when end-user visit it. However, those systems suffer from run-time overhead. Besides, depending on the nature of the attack, clients may have already been exposed to the threats of such malicious websites since the contents have been downloaded before the analysis begins.

3. LITERATURE SURVEY

This section presents a review of the existing methods, tools, and techniques that have been used for phishing detection. The presented studies and research works have used both traditional and modern machine learning approach for detection of phishing websites.

3.1 Existing System

Table -1 below presents a list of currently available phishing plugins together with the techniques they employ, their level of effectiveness, and service type. Each of the plugins was developed for a specific browser, and not all are built for a cross-platform application. Thus there is an inherent weakness in the build of many plugins because end-users may have to use a browser that they are not used to when accessing content on the Internet and this reduces their efficacy. Also, the existing systems are trained and tested on a dataset of much low volume and thus the performance achieved and the presented output statistics may seem doubtful.

Table-1: Existing Systems

SYSTEM	BROWSER	FEATURES				ALGORITHM	ACCURACY (%)
		BLACK LIST	LEXICAL	HEURISTIC	VISUAL		
GoldPhish	Internet Explorer	No	Yes	Yes	Yes	Google PageRank	98
Google Safe Browsing	Chrome/ Firefox	Yes	No	No	No	Google PageRank	93.3
Microsoft SmartScreen	Internet Explorer	Yes	Yes	No	No	Matching	95.9
Cloudmark	Internet Explorer	No	Yes	Yes	No	Matching	94
SpoofGuard	Internet Explorer	No	Yes	Yes	No	Image hash	91
PhishDef	Chrome	No	No	Yes	No	Support Vector Machine	97
Cantina+	Internet Explorer	No	Yes	Yes	No	TF-IDF	98.06
PhishAri	Chrome	No	Yes	No	No	Random Forest	97.52
PhishZoo	Chrome	No	Yes	Yes	No	Fuzzy hashing	96.10
Phishdentity	Explorer	No	Yes	Yes	Yes	image API	97.2

3.2 Related Work

The works presented below are the research works conducted in the phishing website detection domain using the detection technique of applying advanced machine learning and deep learning algorithms and techniques. The techniques involved in these research works yielded results that showed better performance as compared to the traditional phishing detection techniques

In this paper, the authors presented a tool with a collection of hybrid classifier features to detect URLs using on machine learning algorithms. The main feature set is extracted using the cumulative distribution gradient technique, while the info perturbation ensemble technique is employed to extract the secondary feature set. The algorithm used for training the classifier is Random Forest in association with ensemble

learners identifies the phishing websites with a precision of 94.6 percent. [1]

With machine learning and deep learning algorithms, the authors made a relative study to detect phishing website URLs. Convolution Neural Network (CNN) and CNN Long STM (CNN-LSTM) with Logistic Regression formed the architecture of the classification model. The system was designed using tools like TensorFlow and Keras for machine learning and the deep learning model. The dataset was imported from multiple sources to supply better scalability. The malicious or spam website URLs were imported from MalwareDomains, while the phishing website URL dataset was obtained from OpenPhish and Phishtank. [2]

The proposed system detected phishing websites employing a machine learning algorithm. The feature set included six features that supported the website structure and were chosen after a comparative study by the authors. To classify websites whether legitimate or phishing the classifier was trained using Support Vector Machine which worked effectively. The model presented obtained an accuracy of 84% for the classification of websites. [3]

In this paper, to detect phishing websites the authors designed a browser extension. The system used multiple machine learning algorithms including Random Forest, Support Vector Machine (SVM), and k-Nearest Neighbor (kNN) to coach the classifier to realize higher precision by doing a comparative study. The feature set included HTML features of the websites and a content-based approach for extracting the JavaScript. To detect phishing websites and boasted a 22 feature classification technique the dataset was imported from UCI-Machine Learning Repository. [4]

Authors made a comparative study of various machine learning algorithms such as Random Forests, Support Vector Machines (SVM), Logistic Regression (LR), Bayesian Additive Regression Trees (BART), and Neural Networks to implement an efficient phishing website detection system. The imported dataset consisted 2889 URLs that were classified as phishing and a set of true blue messages. In total 43 features were extracted from the acquired dataset and were used extensively to coach the classifier using the machine learning algorithms to get higher precision and accuracy. [5]

This paper proposes it reduces feature classification using a phishing website detection method. The extracted features were analyzed using Support Vector Machine (SVM) and Logistic Regression algorithms. Out of the entire 30 features identified, 19 features were selected and used for classification. The model was implemented using Big Data and therefore the Dataset was obtained from the UCI Irvine machine learning repository. Between the two algorithms used, Support Vector Machine (SVM) showed better performance and accuracy of 95.62%. [6]

The authors designed a system with a detection technique involving a fresh approach for phishing website detection named PhishLimiter. The proposed system used Deep Packet Inspection (DPI) along with side Software-Defined Networking (SDN) through web communications and emails for identifying malicious activities. The real-time DPI and phishing signature classification supported SDN programmability provided PhishLimiter, the pliability to deal with phishing attacks in real-time.

The real world environment attacks were evaluated with the help of efficient management in network traffic proved a better solution to detect phishing websites. [7]

The authors during this paper proposed a phishing detection system with a feature classification methodology. From Google and Phishtank the phishing and legitimate website URL datasets were imported. Using the consistency subset-based feature selection methods and the WEKA tool 133 features were extracted from the obtained dataset. Algorithms like Naïve Bayes and Sequential Minimal Optimization were used to train the classifier to detect the phishing website URL. After doing a comparative study, the authors concluded that using Naïve Bayes in terms of detecting the websites Sequential Minimal Optimization (SMO) achieved better performance. [8]

In this paper, the authors used AI techniques like neural networks for detecting phishing websites. The obtained data set from third-party service providers was divide into two parts, each for a selected purpose. The training module used 20 percent dataset for the Testing phase while it imports 80 percent of the dataset. The Neural network model utilized the input of 17 neurons to match with 17 characteristics within the imported dataset. The system determined whether the website is legitimate or phishing supported one hidden layer level of processing and output of two neurons. The proposed system showed an accuracy of 92.48.

The authors in this paper presented a system to detect URL as benign or phishing. The model was implemented in MATLAB and therefore the Data Set was imported from the UCI Irvine machine learning repository. The system comprises of extraction of features from websites using Extreme Learning Machine (ELM), Naïve Bayes, and SVM. The Extreme Learning Machine (ELM) produced an accuracy of 95.34% among all the applied classifiers. The model was implemented in MATLAB and therefore the Data Set was imported from the UCI repository. [10]

4. Problem Statement

Phishing URL classification and detection suffers from the problem unpredictable and unstable detection criteria including numerous classification components. Many conventional tools and techniques for classifying phishing URLs are suggested to deal with this issue. However,

detecting phishing websites is a challenging task, as most of these techniques make an inaccurate decision. Registering a new domain has become easier, hence no comprehensive blacklist can ensure a perfect up-to-date database. Furthermore, page content inspection has been employed by some strategies to beat the false-negative problems and complement the vulnerabilities of the stale lists. Moreover, each page content inspection algorithms have a different approach to overcome the false-negative problems and complement the vulnerabilities of the stale lists with varying degrees of accuracy.

5. OBJECTIVES

After analyzing all the existing systems and research works presented in the phishing website detection domain we have identified the drawbacks of these systems and the objectives presented below can help design a system with better reliability and performance.

- To develop an effective detection tool that tracks and discovers malicious pages.
- To identify phishing websites employing a combined approach by constructing resource description framework models and using machine learning and group learning algorithms to classify websites.
- To develop a system that can slot into an existing classification system, receiving a URL submitted through an Application Programming Interface and determining whether it is malicious or not
- To classify incomplete data sets as well as eliminate features that allow quantifying the frequency of each task within the dataset.
- To achieve at least a minimum performance of around the rate of a human classification typically around one every ten seconds, but hopefully, we will be able to significantly improve on this rate.

6. DETECTION TECHNIQUE

Phishers usually distort the hostname part and the path part from the URL of the target web page to produce the phishing URL, and thus features can be extracted based on the URL statistical rules or simply based on the URL strings. Studies have proposed many interesting features of various styles of phishing websites from multiple perspectives. To outline a framework that can give assurance from phishing attacks, there exist various ways. Several techniques and methodologies are being applied in the frameworks presented before. Anti-Phishing solutions can be classified into the following categories:-

a) Blacklist based approach
A blacklist includes an inventory of internet sites that are declared as spam. Such blacklists are maintained by organizations like Google. Spam URLs are added to this blacklist. The disadvantage of this approach is that a newly created phishing URL may not be present on the blacklist. Thus, such URLs will be left undetected. URLs present in the blacklist have been denied access. This suggests a user cannot browse this webpage.

b) Heuristic-based approach
To classify URLs this technique makes use of heuristics. Heuristics are the features that are considered to check a website. Here the heuristics like IP address in domain part, '@' symbol in URL, right-click disabled, pop-up windows for passwords, etc. to derive rules on these heuristics and choose a threshold for it.

c) Content-based approach
The comparison of two web pages is done based on the similar contents on the web page. This technique of Term Frequency/Inverse Document Frequency (TF-IDF) is used in this approach. TF-IDF compares the terms in the original website to the phish one. Another approach is to capture the screenshots of a website and then process them to compare. This information retrieved from screenshots after processing can be given to a search engine to acquire its page rank and check the legitimacy of the website by comparing the content on it.

d) Machine learning-based approach
In this technique, features are extracted, and that they are classified using machine learning techniques. The classification performance depends on the algorithm applied. We can see that more than one ML method is experimented on the same dataset to find the best suitable one. Such comparisons of algorithms can help to offer better accuracy in experimentation.

7. PROPOSED METHODOLOGY

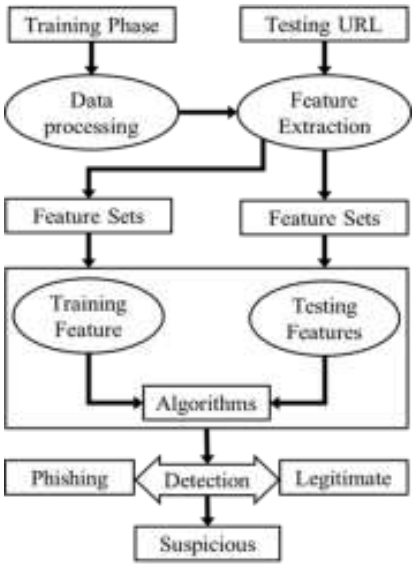


Fig -4: System Design Flow

Our design approach is divided into 2 phases:

a) Training phase:
This phase includes importing and acquiring the dataset, preprocessing the dataset of phishing and legitimate

website URLs, extracting various features from the processed dataset, and applying machine learning algorithms to train the classifier with the help of extracted feature attributes.

b) Testing Phase:
This phase tests the trained classifier with website URLs in a real-time environment and determines whether the website is phishing or benign.

Advantages of the proposed methodology

- Easily identifies trends and patterns
  Machine Learning can review large volumes of data and determine specific trends and patterns that would not be apparent to humans.
- No human intervention needed (automation)
  With ML, you are doing not need to babysit your project every step of the way. Since it means giving machines the facility to seek out, it lets them make predictions and also improve the algorithms on their own. ML is additionally good at recognizing spam.
- Continuous Improvement
  As algorithms of ML gain experience, they keep improving in efficiency and accuracy. This lets them make better decisions. Say you'd wish to form a weather forecast model. Because the amount of data you've keeps growing, your algorithms learn to make more accurate predictions faster.
- Handling multi-dimensional and multi-variety data
  Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, which they will do in dynamic or uncertain environments.

8. DATASET

For all machine learning techniques, collections of data which is termed as a dataset is a crucial and important step for classification purposes. The type and size of the dataset influence the prediction of such detection systems to a great extent. The resulting prediction model performs the best when it matches the real-time data. For our dataset, the URLs were imported from three different sources as follows:

- Alexa: In total, 10 million top website URLs were imported and stored as legitimate URLs dataset
- Phishtank: It is a repository of phishing URLs which is the source of our phishing URLs dataset
- UCI Machine Learning Repository: It has a set of databases spanning various categories and the phishing dataset was imported for our training and testing dataset.

The imported URLs was merged in a single dataset and then a sample of 11055 URL was created for evaluation purposes. Dataset splitting was applied to the sampled dataset to create two datasets for training and testing classifiers.

9. FEATURE REPRESENTATION

As stated earlier, the success of a machine learning model critically depends on the standard of the training data, which hinges on the standard of feature representation.

The process of feature representation can be further broken down into two steps:

a) Feature Collection: This phase is engineering-oriented, which aims to collect relevant information about the URL. This includes information such as the presence of the URLs in a blacklist, features obtained from the URL String, information about the host, the content of the website such as HTML and JavaScript, popularity information, etc.

b) Feature Preprocessing: In this phase, the unstructured information about the URL (e.g. textual description) is appropriately formatted and converted to a numerical vector so that it is often fed into machine learning algorithms.

Table-2: Extracted Features Attributes

Sr. No.	Feature Attribute	Attribute Values	Feature Category
1	Using the IP Address	{ -1,1 }	Address Bar based Features
2	Long URL	{ 1,0,-1 }	
3	URL Shortening Services	{ 1,-1 }	
4	having “@” Symbol	{ 1,-1 }	
5	Redirecting using “//”	{ -1,1 }	
6	Adding Prefix or Suffix	{ -1,1 }	
7	Multi Sub Domains	{ -1,1,0 }	
8	SSLfinal_State	{ -1,1,0 }	
9	Domain_Reg_Length	{ -1,1 }	
10	Favicon	{ 1,-1 }	
11	Using Non-Standard Port	{ 1,-1 }	
12	HTTPS” Token	{ -1,1 }	
13	Request URL	{ 1,-1 }	Abnormal Based Features
14	URL of Anchor	{ -1,0,1 }	
15	Links in tags	{ 1,-1,0 }	
16	Server Form Handler	{ -1,1,0 }	
17	Information to Email	{ -1,1 }	
18	Abnormal URL	{ -1,1 }	
19	Website Forwarding	{ 0,1 }	HTML and JavaScript based Features
20	Status Bar Customization	{ 1,-1 }	
21	Disabling Right Click	{ 1,-1 }	
22	Using Pop-up Window	{ 1,-1 }	
23	IFrame Redirection	{ 1,-1 }	
24	Age of Domain	{ -1,1 }	Domain based Features
25	DNS Record	{ -1,1 }	
26	Website Traffic	{ -1,0,1 }	
27	PageRank	{ -1,1 }	
28	Google Index	{ 1,-1 }	
29	Links Pointing to Page	{ 1,0,-1 }	
30	Statistical-Reports	{ -1,1 }	

10. CLASSIFIER ANALYSIS

The presented classifiers are various advanced machine learning and deep learning algorithms. The algorithms showing better precision and higher accuracy in the phishing website detection systems are mentioned below:

10.1 Logistic Regression

Logistic Regression is used to predict the probability of categorical dependent variables. It takes in binary input where data is coded as 1(true) and 0(false). This algorithm is simple to use and implement and also compatible with the dataset. The algorithm complements the dataset as one of the drawbacks of this algorithm is that it performs poorly if complex non-linear relationships exist between the variables. Logistic Regression is easy to implement yet provides great training efficiency in some cases and it is one of the simplest machine learning algorithms. Logistic regression is less prone to over-fitting. As this algorithm is sensitive to outliers the dataset may lead to incorrect results.

10.2 AdaBoost

The first successful boosting algorithm designed for binary classification was AdaBoost. To increase the performance of decision trees on binary classification problems AdaBoost is best. The authors of the technique Freund and Schapire originally named AdaBoost as AdaBoost.M1. To use for classification more than regression AdaBoost is used. To boost the performance of any machine learning algorithm AdaBoost can be used. This model achieves accuracy just above random chance on a classification problem. Decision trees with one level are the most common algorithm used with AdaBoost. They are often called decision stumps because these trees are so short and only contain one decision for classification. With less need for tweaking parameters AdaBoost is easier to use than SVM. To improve the accuracy of your weak classifiers and making them flexible AdaBoost can be used. If you plan to use AdaBoost then it is highly recommended to eliminate noisy data because it is extremely sensitive.

10.3 K-Nearest Neighbor

The k-nearest neighbor algorithm may be an easy, simple-to-implement supervised machine learning algorithm that will not be solved by both regression and classification problems. The KNN algorithm assumes that similar objects are near to each other. In other words, similar things exist nearby. There's no need to build a tune several parameters, model, or make additional assumptions. The algorithm is versatile. They are often used for classification, regression, and search. As the number of examples or predictors or independent variables increases the algorithm gets significantly slower.

10.4 Bagging Classifier

To aggregate their predictions to form a final prediction a bagging classifier is used. Each classifier is trained with a training set which is generated by randomly drawing, with replacement. Bagging is a completely data-specific algorithm. This technique reduces model over-fitting. On high-dimensional data, it performs well. Based on the mean predictions from the subset trees it gives its final prediction, rather than outputting the precise values for the classification or regression model.

10.5 Decision Tree

To represent decisions and decision making visually and explicitly decision tree can be used. This methodology is more commonly referred to as learning decision tree from data and the above tree is named Classification tree because the target is to classify passenger as survived or dead. Regression trees are represented within the same manner, just they predict continuous values like the price of a house. Decision trees can generate understandable rules. It gives a clear indication of which fields are most important for classification or prediction. For estimation tasks, decision trees are less appropriate where the goal is to predict the value of a continuous attribute. A decision tree can be expensive to train.

10.6 XGBoost

To push the limit of computations resources for boosted tree algorithms XGBoost is used. It is extremely fast and highly efficient. It is versatile. It can be used for regression, classification, or ranking. Can be used to extract variable importance. It does not require missing values imputation, scaling, and normalization. It can only work with numeric features. If hyperparameters are not tuned properly it leads to overfitting.

Specifically, XGBoost supports the following main interfaces:

- Command Line Interface.
- C++.
- Python interface
- R interface.
- Julia.

10.7 Gradient Boosting

Gradient boosting, a bit like the other ensemble machine learning procedure, sequentially adds predictors to the ensemble and follows the sequence in correcting preceding predictors to reach an accurate predictor at the end of the procedure. To pinpoint the challenges within the learners' predictions used previously gradient boosting utilizes gradient descent. By combining one weak learner with the

next learner, the error is reduced significantly over time due to this error is highlighted. Often provides predictive accuracy that can be trumped. Lots of flexibility. Pre-processing of data is not required. Handles missing data - imputation not required. Gradient Boosting Models will continue improving to attenuate all errors. This will overemphasize outliers and cause overfitting. Computationally expensive - often require many trees which may be time and memory exhaustive. This needs an outsized grid search during tuning. Less interpretative in nature, although this is often easily addressed with various tools.

10.8 Gaussian Naive Bayes

It is a variant of Naive Bayes that follows Gaussian distribution and supports continuous data.
When working with continuous data, an assumption often taken is that the continual values related to each class are distributed consistent with a traditional (or Gaussian) distribution.

Sometimes assume variance

- is independent of Y (i.e., σi)
- or independent of Xi (i.e., σk)
- or both (i.e., σ)

The continuous-valued features and models each as conforming to a Gaussian distribution and Gaussian Naive Bayes supports it. An approach to making an easy model is to assume that the info is described by a normal distribution with no co-variance (independent dimensions) between dimensions. This algorithm saves a lot of time and works quickly. Naive Bayes is suitable for solving multi-class prediction problems. If it's the assumption of its independence features is true, it can perform better than other models and requires much less training data. It suits better for categorical input variables than numerical variables. It assumes that all features are independent in real life but there are limits to the applicability of this algorithm in real-world use cases.

10.9 Random Forests

Random Forests uses methodologies of classification and regression with multiple classifier algorithms. It constructs a decision tree at training time to predict possible consequences. Random forest is an ensemble algorithm based on Bootstrap Aggregation (bagging technique), that creates a set of decision trees on randomly multiple samples of the training set, gets a prediction from each tree, and, employing voting of these trees results, gives a better estimation for the final class of the test object. In its approach, instead of gets optimal split points for trees, by the randomness of the selected subset of the training set, it selects suboptimal splits. Due to this, different models will be created, which will be aggregated by combining their results. It reduces overfitting in decision trees and helps to improve the accuracy. It is flexible to both classification and regression problems. It works well with both categorical and continuous values. It automates missing values present in the data.

10.10 Support Vector Machine

Support Vector Machines commonly known as SVMs are used for both regression and classification purposes. Support Vector Machines (SVM) refers to a supervised learning algorithm, in which the objective is to find a hyperplane in the input variable space to best separate the data points into two classes. This choice is based on that hyperplane that has the most significant margin, which is that hyperplane that presents the maximum distance between data points of both classes. By doing this, new data points can be sorted with more accuracy and precision. Those points that are closer to the hyperplane are named Support Vectors. They influence the position and orientation of the hyperplane, as well as the number of features that influence the dimension of the hyperplane. SVM is more effective in high-dimensional spaces.

11. EVALUATION METRICS

To evaluate the performance of our presented model, the parameters of the most commonly used evaluation metrics that are Accuracy, Precision, True Positive Rate (TPR), and False Positive Rate (FPR) were used for performance comparison of the classifiers. Accuracy, Precision and TPR, FPR are defined as follows:

$$TPR = \frac{Phishing_{phish}}{Phishing_{phish} + Phishing_{legit}}$$

$$FPR = \frac{Legitimate_{phish}}{Legitimate_{phish} + Legitimate_{legit}}$$

$$Precision = \frac{Phishing_{phish}}{Phishing_{phish} + Legitimate_{phish}}$$

$$Accuracy = \frac{Phishing_{phish} + Legitimate_{legit}}{Phishing + Legitimate}$$

where,
Phishingphish represents the total no. of Phishing URLs correctly classified as Phishing URLs

Phishinglegit represents the total no. of Phishing URLs wrongly classified as Legitimate URLs,

Legitimatelegit represents the total no. of Legitimate URLs correctly classified as Legitimate URLs

Legitimatephish represents the total no. of Legitimate URLs wrongly classified as Phishing URLs.

Phishing represents the total no. of Phishing URLs.
Legitimate represents the total no. of legitimate URLs.

12. CLASSIFIER TRAINING

The proposed detection algorithm works as detecting phishing websites from the selected datasets. Firstly the user selects the dataset to process the data. Secondly, all the attribute values will be calculated and according to the accuracy, the values are shown. This algorithm calculates the attributes and values explained in the performance evaluation. Classifier Training Algorithm:

1) Import and Preprocess Dataset.
   - Blacklist and Whitelist of URLs
   - Phishing and Benign Websites
2) Extract the features of the URL
   - The address bar and Domain name features
   - HTML and JavaScript features
3) Compute attribute values, if
   - Attribute present value = 1
   - Attribute absent value = -1
   - Attribute not considered = 0
4) Select and Input attributes
   - Attribute X (feature 1 of an URL)
   - Attribute Y (feature 2 of an URL)
   - Calculate threshold value
5) Find Range Value to get interlinked value
   - A = threshold value of attribute X
   - B = threshold value of attribute y
6) Select Attribute to get Threshold Value
   - Threshold value = 1 (feature present)
   - Threshold value = -1 (feature absent)
7) Classify Websites URLs using Threshold Value
   - Feature present = Phishing URL
   - Feature absent = Legitimate URL
8) Compute Performance
   - Sensitivity (True Positive Rate)
   - Specificity (True Negative Rate)

The threshold value and range value will be identified using the attribute values extracted and computed from the extraction of the URL features. The values for each phishing attribute is ranging from {-1, 0, 1} these values are defined as low, medium, and high according to the phishing website feature. The classification of phishing and legitimate URLs is based on the values of attributes extracted using four types of phishing categories and a machine learning approach.

13. COMPARITIVE ANALYSIS

In total 10 machine learning classifiers were applied to the training and testing dataset obtained on splitting the sample dataset in the ratio 60:40. Advanced machine classifiers like Random Forest, Bagging Classifier, Decision Tree, Gradient Boosting, XGBoost, Support Vector Machine, k-Nearest Neighbor, Logistic Regression, AdaBoost, and Gaussian Naïve Bayes used for the training and testing purposes and the results of each classifier were empirically compared with the evaluation metrics such as accuracy, precision, recall and f1 score obtained for both training and testing respectively.

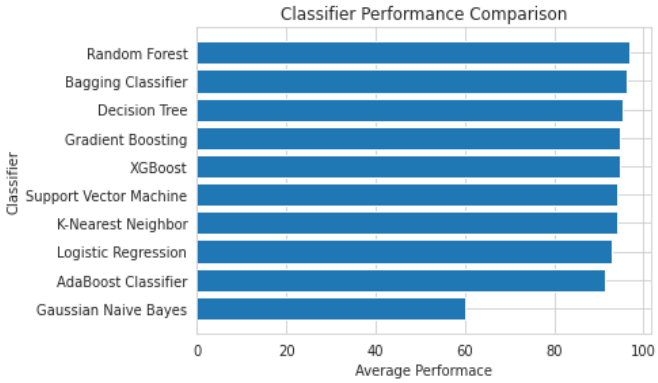


Fig -5: Classifier Performance Graph

The above figure shows a comparison of the performance of the classifier based on the parameter of accuracy obtained while the testing phase of the system.

Table-3: Existing Systems

Sr. No.	Classifier	Evaluation Parameters			
		Accuracy	Precision	Recall	f1 Score
1.	Random Forest	98%	99%	99%	99%
2.	Bagging Classifier	98%	99%	96%	96%
3.	Decision Tree	98%	99%	95%	95%
4	Gradient Boosting	94%	95%	95%	95%
5.	XGBoost	94%	95%	95%	95%
6.	Support Vector Machine	94%	94%	94%	94%
7.	k-Nearest Neighbor	94%	94%	94%	94%

8.	Logistic Regression	92%	93%	93%	93%
9.	AdaBoost	91%	91%	91%	91%
10.	Gaussian Naïve Bayes	59%	88%	60%	65%

As shown in the above table, the Random forest classifier performed best considering all the performance measures. Hence for the final prediction model, a random forest classifier is chosen to yield the best performance in terms of accuracy and precision for the extension/plugin.

14. FINAL PREDICTION MODULE

The system is overall split into backend and plugin. The backend consists of dataset preprocessing and training modules. The frontend which is the plugin consists of JavaScript files for content script and background script including the Random Forest script. The plugin also consists of HTML and CSS files for the user interface. This module takes the feature vector from the feature extraction module and the JSON format from the Exporting model module and then gives a Boolean output that denotes whether the webpage is legitimate or phishing. The plugin is distributed as a single file and requires a Chrome browser to run. The plugin (frontend) is packed into a .crx file for distribution.

15. RESULTS SNAPSHOTS

The Phishing Detector Plugin is unpacked as an extension in the Google Chrome Browser. The plugin dashboard contains two functions, one is to check the active URL in the browser and classify it as phishing or legitimate and the second function enables the user to scan a suspicious QR code to know its content

Fig -6: Phishing Detector Plugin Dashboard

The QR code scanner integrated into the plugin helps the user to know the contents of the QR code before accessing it.

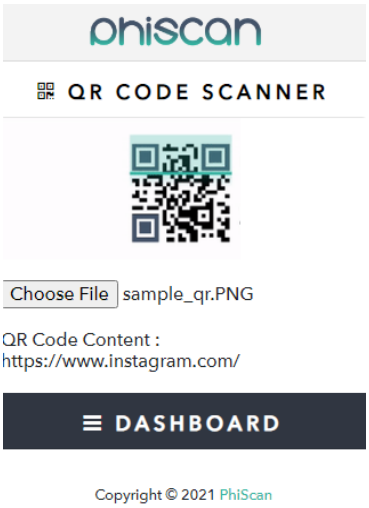


Fig -7: QR Code Content Scanner

The plugin checks the active URL accessed by the user and if the URL is legitimate then no action or alert is shown to the user. The user can still click on the extension button to know the details of the accessed URL.

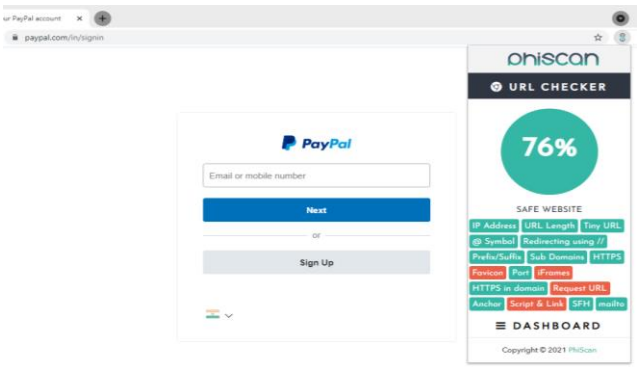


Fig -8: Plugin correctly identified the legitimate website of PayPal

If the URL accessed by the user is classified as phishing by the plugin, then an alert box appears with a message "Warning!!! Suspicious Website is being loaded". The user can click on the extension button to know the details of the accessed webpage

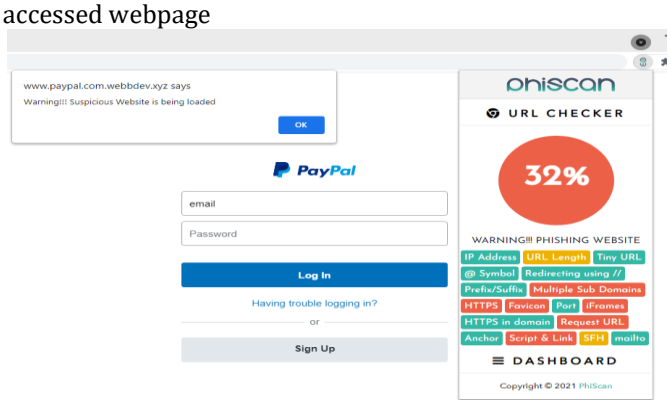


Fig -9: Plugin successfully identified a phishing website imitating the PayPal login webpage

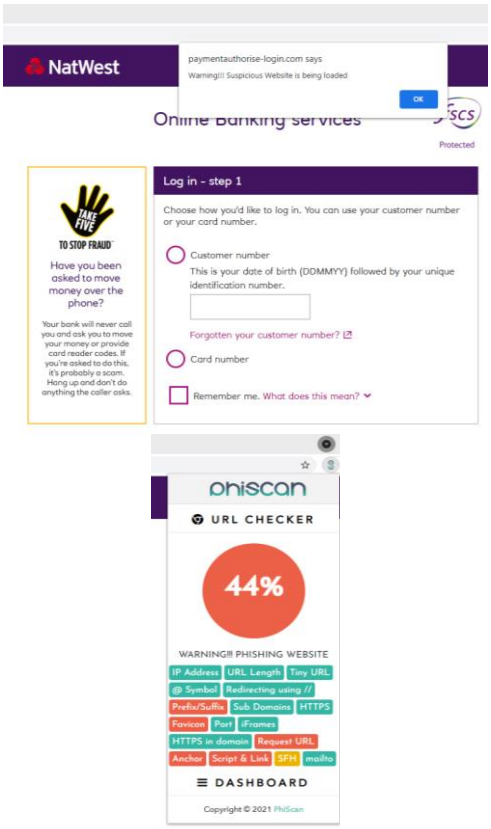


Fig -10: Plugin successfully identified a phishing website imitating the NatWest website

16. CONCLUSION

This is a phishing website detection system that focuses on client-side implementation with rapid detection so that the users will be warned before getting phished. The main implementation is porting of Random Forest classifier and Decision Tree to JavaScript. Similar works often use webpage features that are not feasible to extract on the client-side and this results in the detection being dependent on the network. On the other side, to able to provide rapid detection and better privacy this system uses only features that are possible to extract on the client-side. Although using lesser features results in a mild drop inaccuracy, it increases the usability of the system. This work has identified a subset of webpage features that can be implemented on the client-side without much affecting accuracy. As the JSON representation of the model and the classification script is designed with time complexity in mind the port from python to JavaScript and own implementation of Random Forest and Decision Tree in JavaScript further helped in rapid detection. The plugin can detect phishing even before the page loads completely. The F1 score calculated on the test set on the client-side is 88%. A lot of improvements and enhancements are possible in this domain of phishing website detection systems. The main aim of the system is to alert the user immediately on accessing the phishing or suspicious URL and the presented detection plugin is successful in doing the same while maintaining high accuracy.

REFERENCES

[1] Kang Leng Chiew, Choon Lin Tan, KokSheik Wong, Kelvin SC Yong, and Wei King Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," Information Sciences, vol. 484, pp. 153-166, 2019

[2] A. Vazhayil, R. Vinayakumar, and K. Soman, "Comparative Study of the Detection of Malicious URLs Using Shallow and Deep Networks," in 2018 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018, 2018, pp. 1–6.

[3] Pan, Ying, and Xuhua Ding. ―Anomaly-based web phishing page detection.‖ In Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual, pp. 381-392. IEEE, 2006.

[4] A. Desai, J. Jatakia, R. Naik, and N. Raul, "Malicious web content detection using machine learning," RTEICT 2017 - 2nd IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. Proc., vol. 2018–Janua, pp. 1432–1436, 2018.

[5] Abu-Nimeh, Saeed, Dario Nappa, Xinlei Wang, and Suku Nair. "An examination of machine learning systems for phishing recognition." In Proceedings of the counter phishing working gatherings second yearly eCrime specialists summit, ACM, pp. 60-69, 2007.

[6] W. Fadheel, M. Abusharkh, and I. Abdel-Qader, "On Feature Selection for the Prediction of Phishing Websites," 2017 IEEE 15th Intl Conf Dependable, Auton. Secur. Comput. 15th Intl Conf Pervasive Intell. Comput. 3rd Intl Conf Big Data Intell. Comput. Cyber Sci. Technol. Congr., pp. 871–876, 2017.

[7] Tommy Chin, Kaiqi Xiong, and Chengbin Hu, "PhishLimiter: A Phishing Detection and Mitigation Approach Using Software-Defined Networking", IEEE Access, 2018.

[8] M. Aydin and N. Baykal, "Feature extraction and classification phishing websites based on URL," 2015 IEEE Conf. Commun. Network security, CNS 2015, pp. 769–770, 2015.

[9] Rami M Mohammad, Fadi Thabtah, and Lee McCluskey, "Predicting phishing websites based on self-structuring neural network," Neural Computing and Applications, vol. 25, pp. 443-458, 2014.

[10] Y. Sönmez, T. Tuncer, H. Gökal, and E. Avci, "Phishing web sites features classification based on extreme learning machine," 6th Int. Symp. Digit. Forensics Secur. ISDFS 2018 - Proceeding, vol. 2018–Janua, pp. 1–5, 2018.

[11] Xiang, Guang, and Jason I. Hong. ―A hybrid phish detection approach by identity discovery and keywords retrieval.‖ In Proceedings of the 18th international conference on World Wide Web, pp. ACM, 2009.

[12] L. Machado and J. Gadge, "Phishing Sites Detection Based on C4.5 Decision Tree Algorithm," in 2017 International Conference on Computing, Communication, Control, and Automation, ICCUBEA 2017, 2018, pp.

[13] Meena, p., m. Kavitha, s. jeyanthi, and cpnijithamahalakshmi. "Phishing prevention using data mining techniques." International Journal of Pure and Applied Mathematics 119, no. 10 117-123, 2018.

[14] M. Karabatak and T. Mustafa, "Performance comparison of classifiers on reduced phishing website dataset," 6th Int. Symp. Digit. Forensic Secur. ISDFS 2018 - Proceeding, vol. 2018–Janua, pp. 1–5, 2018.

[15] Peng Yang, Guangzhen Zhao, Peng Zeng, "Phishing Website Detection based on Multidimensional Features driven by Deep Learning", IEEE Access, 2018.

[16] S. Parekh, D. Parikh, S. Kotak, and P. S. Sankhe, "A New Method for Detection of Phishing Websites: URL Detection," in 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, vol. 0, no, pp. 949–952.

[17] Solomon Ogbomon Uwagbole, William J Buchanan, and Lu Fan, "Applied machine learning predictive analytics to SQL injection attack detection and prevention," in 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 2017, pp. 1087- 1090.

[18] K. Shima et al., "Classification of URL bitstreams using a bag of bytes," in 2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), 2018, vol. 91, pp. 1–5.