URL Phishing Detection using Machine Learning Technique

MSc Research Project
Cyber Security

Naveen Kumar Singh
Student ID: x19223978

School of Computing
National College of Ireland

Supervisor: Niall Heffernan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Naveen Kumar Singh
Student ID:	X19223978
Programme:	MSc in Cyber Security **Year:** 2020-21
Module:	Research Project
Supervisor:	Niall Heffernan
Submission Due Date:	16/08/2021
Project Title:	URL Phishing Detection using Machine Learning Technique
Word Count:	5881 **Page Count** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Naveen Kumar Singh

Date: 15/08/2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

URL Phishing Detection using Machine Learning Technique

Naveen Kumar Singh
X19223978

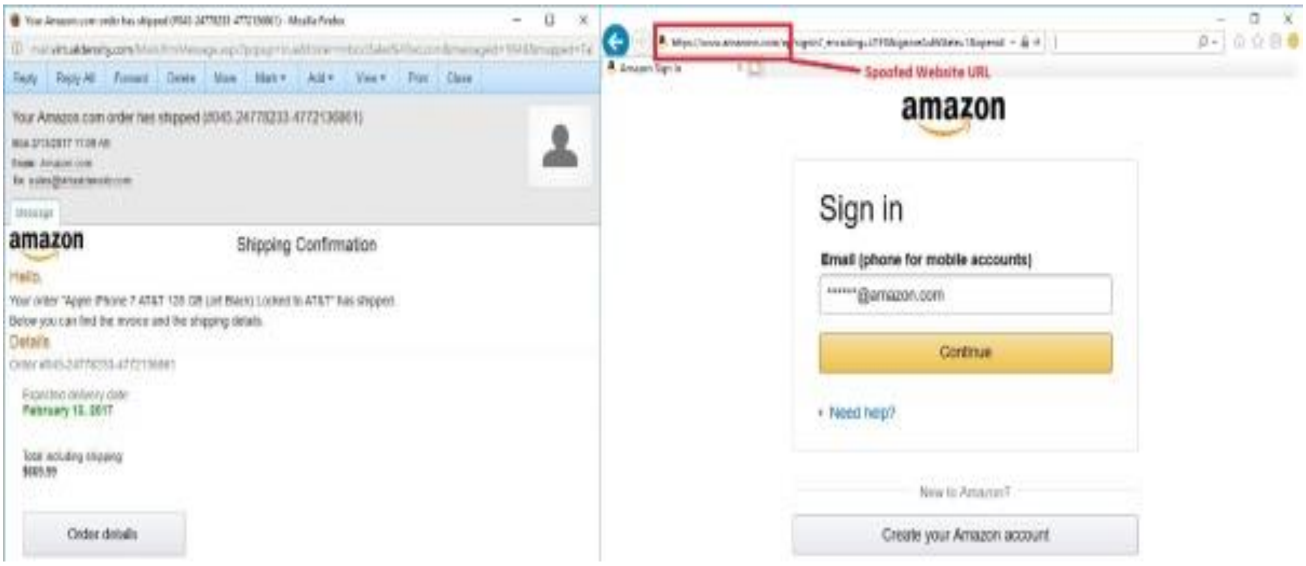
Abstract

One of the primary worries of security researchers nowadays is the staggering number of phishing attempts. Traditional phishing website detection technologies rely on signature-based techniques that are incapable of detecting recently generated phishing websites. As a result, researchers are developing Machine Learning-based algorithms capable of detecting and classifying phishing websites with high degree of accuracy when a vast number of characteristics are evaluated. Building a classification model with a vast number of characteristics, on the other hand, requires time, which impedes the rapid recognition of phishing websites. As a result, it is important to use a feature selection approach to shortlist a collection of features so that high-performance classification models may be constructed in less time. The performance of Machine Learning methods with and without feature selection is compared. Experiments are carried out on a phishing dataset with 30 characteristics, which includes 4898 phishing and 6157 legitimate websites. According to the comparison findings of the applied classification algorithms, the Random Forest(RF) algorithm performs the best at detecting phishing URLs, with a 91.19 percent accuracy rate.

Keywords - Phishing Detection, URL, Chrome Extension, Machine Learning, Random Forest(RF), Support Vector Machine(SVM), Artificial Neural Networks(ANN)

1 Introduction

People are turning to online platforms instead of visiting banks, stores, and other physical locations due to the low cost of Internet access. Attackers are exploiting this reality by attempting to locate their victims online in order to gain money rather than risking their lives by robbing banks/shops, etc. One of the most serious security concerns on the Internet nowadays is phishing attacks. Attackers frequently use bogus websites to acquire sensitive information from victims. According to the incidents reported to the Federal Bureau of Investigation[19], a theft of around $48 million happened in the United States in 2018. When visiting this sort of fake site, which seems to be the real website, computer users can readily provide critical information without hesitation. Because the inputted webpage appears to be identical to the original webpage.



Your Amazon.com order has shipped (#045-24778233-4772136861)

From: Amazon.com

amazon Shipping Confirmation

Hello,

Your order "Apple iPhone 7 AT&T 128 GB (Jet Black) Locked to AT&T" has shipped.

Below you can find the invoice and the shipping details.

Details

Expected delivery date:
February 13, 2017

Order details

Sign in

Email (phone for mobile accounts)

******@amazon.com

New to Amazon?

Figure 1: Phishing Webpage and mail sample.

In recent years, several techniques and technologies have been created to counteract phishing attempts. These include using heuristics[20],[21] to detect strange websites and educating and training people to prevent phishing attempts[22]. building blacklists[23] and whitelists[24],[25], screening emails containing suspicious URLs, and creating visual signals to distinguish between authentic and fraudulent websites. Furthermore, there is no method for ant phishing specialists to manually evaluate suspicious websites and notify admins to remove the bogus sites. The time lag between publishing a bogus site and removing it is now long enough to entice a lot of victims into disclosing personal information. This scenario drives us to develop a passive testing strategy for detecting phishing websites.

In this work, we present a method for detecting phishing websites through testing. A lot of observations have inspired us. To begin, phishing websites may be thought of as web apps. Phishers change actual online application pages, such as changing a form target URL with his intended URL, in order to get sensitive data. In the process, phishers establish anomalies in application behaviour, such as the acceptance of any random inputs and page navigation, for example, form submission may result in a page with no connections to any other page. Our method is not based on any current whitelists or blacklists. Furthermore, it is unaffected by the language and linguistic content of websites.

The following is how the paper is organized: Section 2 we will examine the literature review conducted in the same study/research field to learn about the perspectives and opinions of other authors. Section 3 is on research methods which will provide us detailed insights and analyses on tools and approaches . Design specifications are covered in section 4. Section 5 goes over the implementation, and Section 6, we will examine the model's performance in depth using the evaluation parameters that we selected while developing the model. Section 7 concludes by drawing some findings and discussing future work.

2 Related Work

This section reviews the literature available in this field under study. We will define the issues and concepts that influences the phishing. The most common type of threat in this modern digital world is Phishing attack which has led to most dangerous cyber-crimes[1]. Phishing is carried by mails and websites only. They both have common goals, but they all use different approaches. To lure users, attackers commit frauds using social engineering schemes such as online advertisements, emails, and messages. The literature review will have 2 sections, in which section 1 include "Traditional Approach vs Machine Learning Approach", in which we will talk about why blacklisting technique is outdated. In section 2, we will discuss about several "Machine Learning Techniques" and why is it important to achieve higher precision and accuracy.

2.1 Traditional Approach vs Machine Learning Approach

Many browsers have built-in phishing detection extensions, for example Google Chrome, Mozilla Firefox based on blacklisted web sites. Traditional approached cannot always stop new evolving websites made for phishing attacks.

That's why we decided to build much more effective built-in browser extension which uses Machine Learning algorithms. Phishers change actual online application pages, such as changing a form target URL with his desired URL, in order to get personal information. In the process, phishers establish discrepancies in application behaviour (i.e., acceptance of any arbitrary input) and page navigation (e.g., A form submission may result in a page with no connections to any other pages.). As a result, their goal is to expose these discrepancies in order to evaluate a suspected web application for phishing. To address these concerns, [2] offer an automated testing technique. Their method complements traditional detection measures aimed towards saving victims. They used a testing technique for suspected phishing websites, addressing such like web apps. The method was tested against 37 reported phishing web apps from 20 various firms. Their method has a decreased false negative rate of less than 3% and a zero false positive rate.

In [3], demonstrated a reliable technique for detecting phishing websites. It uses Machine Learning algorithms for classifications and feature extractions. They compared a total of eight algorithms for feature selection, before and after for classification of websites into legitimate and phishing. The outcome of these experiments is that these algorithms clearly helps detecting phishing websites with much higher accuracy and out of all those algorithms, Random Forest gives the highest accuracy for feature selection.

PhishWHO, a system for detecting phishing websites in three phases, was proposed by [4]. Using N-gram method, the keywords are retrieved from the webpages in the first stage. In second stage, they entered these retrieved keywords into the search engine to retrieve the name of the targeted domain. And finally, they used the 3-Tier Identity Matcher for determining the website's legitimacy.

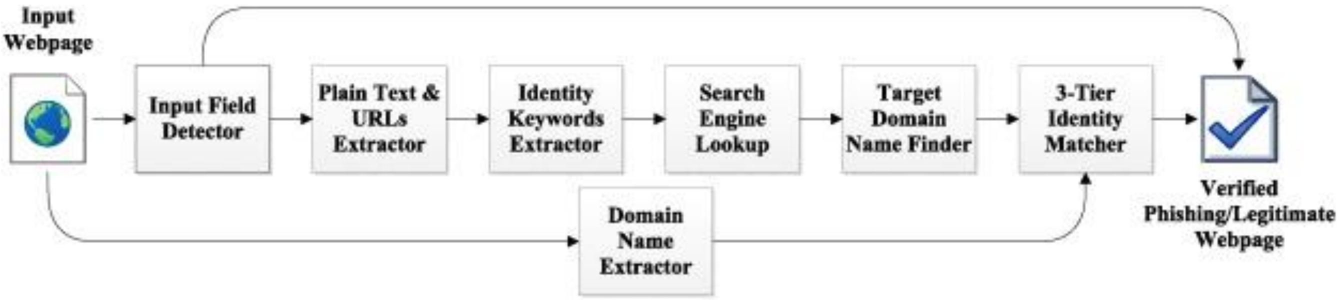

Input Webpage
Input Field Detector
Plain Text & URLs Extractor
Identity Keywords Extractor
Search Engine Lookup
Target Domain Name Finder
3-Tier Identity Matcher
Domain Name Extractor
Verified Phishing/Legitimate Webpage

Figure 2: PhishWHO detector

Blacklist techniques are ineffective in detecting new phishing attacks and have a high false positive rate. Moreover, already invented techniques and systems does not work in real time environment, it is slow and are very much complicated. So, the authors of [5] shortlisted the most essential 19 features after analysing the features of websites. They attained 99.09 percent overall detection accuracy and 99.39% in true positive rate for classification of phishing websites. after using Artificial Neural Network(ANN), Naïve Bayes(NB), Logistic Regression, Random Forest(RF) and Support Vector Machine(SVM).

The authors proposed a technique that combined SVM with a decision tree model[6]. To identify phishing attacks, [6] provide an extension/add-on called PhishDetector. Their experiments even demonstrated that PhishDetector accurately identifies zero-day phishing. They used the support vector machine(SVM) method to categorize webpages using their features vector and to train the model. Then they used decision tree to build the rules for detecting phishing websites. They got the accuracy of 99.14% true positive rate and only 0.86% false negative rate.

[7] presented a URL-based heuristic method. Experiments were done on a dataset including 138 characteristics collected from 16,000 phishing and 31,000 non-phishing URLs. These 138 features are divided into four categories: reputation, statistically mined keyword-based, lexical, and search engine. Here also Random Forest(RF) achieved the highest accuracy among the seven different machine learning algorithms they used.

By integrating the source code and a webpage's URL, [8] exhibited a phishing detection technique. Levenshtein Distance is used as the algorithm for determining string similarity. The machine learning algorithm model in their system was a Support Vector Machine(SVM). They attained 92.6 percent overall detection accuracy.

[9] presented a method for detecting phishing attempts based on human behaviour when exposed to a phony website, which is named FeedPhish. Basically, they feed the fake login credentials through their application into any website before logging in for to check if the website is legitimate or fake. They got this brilliant idea by noticing human behaviour of

some users. In addition to this, they also implemented heuristic filtering to achieve better accuracy. Their application has an overall accuracy of 96.38 percent.

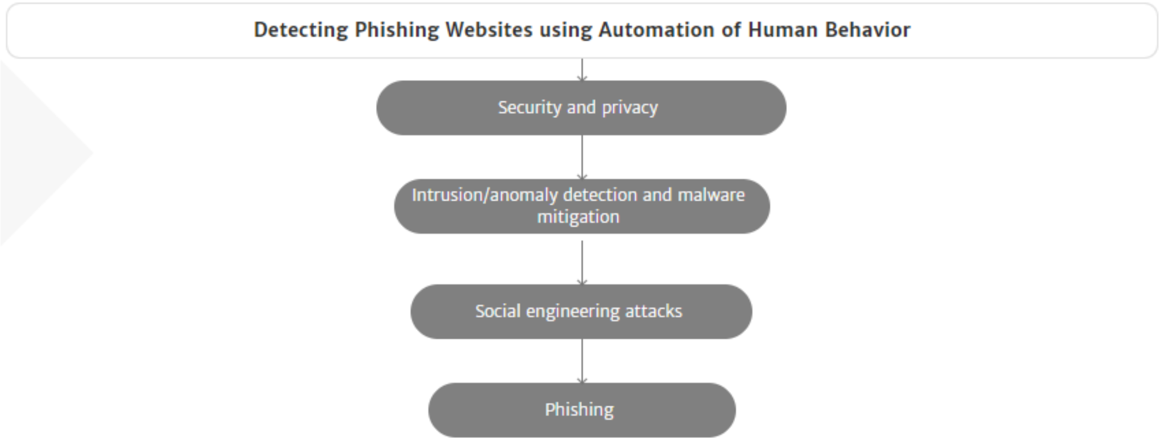

Detecting Phishing Websites using Automation of Human Behavior
Security and privacy
Intrusion/anomaly detection and malware mitigation
Social engineering attacks
Phishing

Figure 3: FeedPhish phishing detector

[10] demonstrated an automated intelligent method for detecting phishing webpages. Using Naïve Bayes algorithm, they initially classified the URL characteristics and then determine whether the page is phishing or legitimate. Then, using SVM, classified its webpage features and determine whether the webpage is phishing or legitimate, if the webpage is suspicious about phishing. Their method has a high detection rate, according to the experimental data.

In [11], authors implemented a phishing detection system using seven distinct Machine Learning algorithms, including K-star, Adaboost, Decision Tree, KNN (n=3), Random Forest, Naive Bayes, and SMO as well as a variety of feature numbers and types, including Natural Language Processing(NLP)-based features, word vectors, and hybrid features. They used their own constructed dataset which has 73,575 URLs(37,175 phishing and 36,400 legitimate URLs). The proposed model has a 97.98 percent accuracy rate.

2.2 Machine Learning Techniques

Machine learning approaches are classified into four types: supervised, unsupervised, semi-supervised, and reinforced. Classifiers for phishing detection that rely on machine learning techniques are classified as follows:

- Random forest(RF) algorithm: Random Forest(RF) algorithm is an intelligent machine learning approach that has lately gained popularity among academics due to its quickness and excellent classification accuracy. In this work done by [12], they used a publicly accessible dataset to train a classifier model for detecting phishing websites in an intelligent and automated manner. According to their findings, Random Forest(RF) is quicker, more robust, and more accurate than the other classifiers. In compared to the other classifiers, the runtime of Random Forest(RF) is relatively quick, and it is capable of detecting phishing websites.

- Neural Network: A neural network is made up of many interconnected neurons. The input layer, the hidden layer, and the yield layer makes up the three layers of neural network. Interconnections are used to transmit messages from one neuron to the next. [13] worked on neural network-based classification using a basic and reliable Monte Carlo method, which does not rely on third parties and has detection stability and enhanced accuracy rate.

- Support Vector Machine(SVM): It is now the most popular and one of the most effective classifiers. It is type of supervised classification method. When there is no knowledge of the data, Support Vector Machine(SVM) performs admirably [14]. A hyperplane or margin divides the two categories in Support Vector Machine(SVM): (phished or legitimate website). When trained on the dataset, the classifier inserts legitimate and phished websites on either plane, therefore categorizing the websites[15].

- K-Nearest neighbour: K-nearest neighbour(KNN) is based on the grouping of items with similar features, it determines the class category of a test case based on its k neighbour [16]. It has been effective accurate results for phishing detection.

- Naïve Bayes Classifiers: Naive bayes is a highlight among the most capable and finest inductive learning computations for Artificial Intelligence and data mining [15].

- Logistic Regression: Logistic regression is mostly used to classify data. The data points in Logistic Regression are not organized in straight rows.

3 Research Methodology

We propose using Machine Learning to address the shortcomings of traditional techniques to phishing detection. The amount of data available to easily find a phishing attack pattern gives advantage to Machine Learning over traditional techniques. The main idea is to apply Machine Learning algorithms on a dataset of phishing websites is to create a model that can be used to determine if a particular website is a legitimate website or a phishing website in real time. We opted to design a Machine Learning model from beginning in JavaScript and use it to create a Chrome extension and later it can be developed into software tool or application. As we open everything or every website on browsers, so it is important to secure the browser from phishing attacks.

For users, the extension/application should not give them wrong results in real world. Therefore, the accuracy of model should be high. For that purpose, the accuracy of trained model should be high also to achieve greater phishing results in testing. When selecting a Machine Learning algorithm for the problem of phishing detection, false positive rates and false negative rates must be taken into account. Because if the false negative rates and false positive rates are high, might show phishing website as legitimate website and vice-versa. The developed method should be able to produce classifications in real time, which means it should utilize less computer resources and have a very short execution time.

3.1 Dataset

We used the ‘Phishing Websites Dataset’ from the UCI Machine Learning Repository[29] to assess our Machine Learning approaches. There are 11,055 URLs (instances) in all, including 4898 genuine/legitimate instances and 6157 phishing instances. Each sample includes 30 website parameters and a class label indicating whether it is a phishing website or not(1 or -1) [17] and 0 denotes that it might be phishing website. For this research implementation, this dataset is used since it is the most recent dataset accessible in the public domain. The following are the classifications of the characteristics represented by the training dataset:

- Using IP (categorical - signed numeric) : { -1,1 }

If IP address is used in URL, then it is stated as phishing website.
Example - http:137.127.5.147fake.html

- Long URL (categorical - signed numeric) : { 1,0,-1 }

Generally, phishing websites use long URLs to hide malicious part of the URL. If the assumption fails, that is, for a genuine webpage with proper lengthy URLs, the omission of other phishing attributes on the website will balance the incorrect assumption and appropriately categorize the genuine webpage as legitimate website.

- Short URL (categorical - signed numeric) : { 1,-1 }

Shortened URLs are usually used to hide the real URL; hence it is suspicious. It is basically use for redirection to real URL. Therefore, it is categorized as phishing website.

- Symbol “@” (categorical - signed numeric) : { 1,-1 }

According to Web standards, the"@" sign is a reserved term. Hence it is suspicious and categorized as phishing website.

- Redirecting “//” (categorical - signed numeric) : { -1,1 }

Redirection of any webpage to any other webpage is represented by “//” symbol. So, to classify any webpage as phishing webpage using this attribute, a rule is established, that is, if the position of "//" in the URL exceeds seven, then it is Phishing webpage.

- Prefix Suffix- (categorical - signed numeric) : { -1,1 }

To give the appearance of a legitimate website, Phishers frequently add a prefix or suffix separated by "-" to the domain of the URL.

- Subdomains (categorical - signed numeric) : { -1,0,1 }

A phishing site is one that includes more than three dots in the domain portion of the URL. Otherwise, it is categorized as legitimate website.

- Server Form Handler (SFH) (categorical - signed numeric) : { 1,0,-1 }

When a form's action handler is blank which looks like "about:blank", or if the domain of the URL differs from the domain of the main URL, it is classified as a phishing site.

- Tags stored in link (categorical - signed numeric) : { 1,-1 }

Tags like <link>, <meta> and <script> contains domain of the link and if it matches the domain in the main URL. Then it is suspicious and categorized as phishing website.

- Request URL (categorical - signed numeric) : { 1,-1 }

External webpage objects such as animations, files, pictures, and so on are often accessible via a request URL that has the same domain as the webpage URL is classified as legitimate website, otherwise it is classified as phishing website.

- Anchor URL (categorical - signed numeric) : { -1,0,1 }

It is considered phishing if the anchor tag simply contains URL fragments or JavaScript functions.

- Submitted information via email (categorical - signed numeric) : { 1,-1 }

If the website has a “mailto:” function, then it is suspicious and categorized as phishing website.

- Status Bar Tampered (categorical - signed numeric) : { 1,-1 }

Phishing or malicious websites can be modified to look like a legitimate website by Phishers using JavaScript. If “onMouseOver” changes the appearance of the status bar, then it is suspicious and categorized as phishing website.

- Favicon (categorical - signed numeric) : { 1,-1 }

If links from other domains are fetched on a particular webpage, then it is suspicious and categorized as phishing website.

- Iframe Redirection (categorical - signed numeric) : { 1,-1 }

To collect user information and redirect to the original website Iframe tags with invisible borders are used by Phishers. If websites that have iframes in the Domain Object Model(DOM) are detected, then it is suspicious and categorized as phishing website.

3.2 Model Training using Machine Learning Algorithms

On the training dataset, we trained and tested supervised machine learning algorithms. The algorithms listed below were considered based on their performance on classification challenges. In a 7:3 split, the dataset was divided into training and testing datasets, which is the most ideal case.

- Random Forest(RF)

Random Forest is a bagging-based integrated learning approach also known as bootstrap aggregating that may be used for classification, regression, and other tasks [18]. In compared to other supervised learning algorithms, it is a fast algorithm since it requires less training time to train. For large datasets, it has a high accuracy. Random Forest has high robustness to outliers and noise and avoids over-fitting; it also has outstanding scalability and parallelism for high-dimensional data classification. Random Forests are classifiers that integrate multiple tree predictors, with each tree based on the values of a randomly chosen vector. Moreover, the forest's trees are all distributed in the same way. We assume that ‘n’ is the number of training observations and ‘p’ is the number of variables that are the features in a training set while creating a tree. We select k « p as the number of variables to be taken to decide the decision node at a tree. In the testing phase, we choose a bootstrap sample from the ‘n’ observations in the training set and use the remaining data to estimate the tree's error. Thus, at each node in the tree, we arbitrarily select ‘k’ variables as a choice and calculate the optimal split based on the ‘k’ variables in the training set. In comparison to other tree algorithms, trees are always growing and never trimmed. Random Forests are capable of handling a high number of variables in a data collection. In addition, they produce an internal neutral approximation of the generalization error during the forest-building process. Furthermore, they are good at estimating missing data. The lack of repeatability is a significant disadvantage of Random Forests, as the process of creating the forest is arbitrary. Furthermore, understanding the final model and subsequent outcomes is challenging since it comprises several separate decision trees.

- Artificial Neural Networks(ANN)

A neural network is made up of linked identical units which is also known as neurons. The interconnections are utilized to transmit signals from one neuron to the next. Furthermore, the interconnections include weights to improve transmission between neurons. The neurons are not particularly strong on their own, but when linked to others, they may conduct complicated calculations. When the network is trained, the weights on the interconnections are changed, so significant interconnections play a larger role during the testing phase. The network is referred to as feedforward since the linkages do not loop back or bypass other neurons. The nonlinearity of the underlying neurons is what gives neural networks their strength. As a result, it is critical to add nonlinearity into the network in order to learn complicated mappings. The sigmoid function is a widely utilized function in neural network research.

- Support Vector Machine(SVM)

Support Vector Machine(SVM) is a supervised learning-based machine learning approach that may be used for both regression and classification. The SVM idea is to build a model for predicting classification based on the presented features of the initial value of the test dataset. The SVM method can use a variety of kernels, although linear, polynomial, RBF, and sigmoid are the most common[28]. A hyperplane divides the two categories in SVM, here it is categorized as phished or legitimate website. The margin is another name for this hyperplane. After training with the dataset, the classifier assigns legitimate and phished websites on either plane, therefore categorizing the websites.

3.3 Model Evaluation

The model's performance will be assessed based on the specific parameters. The confusion matrix will be used to assess the model's efficacy. The matrix is made up of real that is true and false and that is positive and negative values, namely the true positive, true negative, false positive, and false negative.

i. True Positive(TP):
When the predicted value which is positive here turns out to be true, then it is True Positive values.

ii. True Negative(TN):
When the predicted value which is negative here turns out to be true, then it is True Negative values.

iii. False Positive(FP):
When the predicted value which is positive here turns out to be false, then it is False Positive values.

iv. False Negative(FN):
When the predicted value which is negative here turns out to be true, then it is False Negative values.

We can now compute a few evaluation metrics based on the values we have currently, which are as follows.

v. Accuracy:
Accuracy is computed by dividing the total number of correct predictions by the total number of observations in the dataset. The highest accuracy is 1.0, while the lowest is 0.0. (1 – ERR) can also be used to compute it, where ERR is Error rate[26].

Accuracy = (TP + TN) / (TP + TN + FP + FN)

vi. Sensitivity:
The number of correct positive predictions divided by the total number of positives is how sensitivity (SN) is determined. It is also known as the Recall rate (REC) or the True Positive Rate (TPR). Sensitivity ranges from 1.0 to 0.0[26].

Sensitivity = TP / (TP + FN)

vii. Specificity:
The number of correct negative predictions divided by the total number of negatives is used to determine specificity (SP). It is also known as the True Negative Rate (TNR). The highest specificity is 1.0, while the lowest is 0.0.

Specificity = TN / (TN + FP)

4 Design Specification and Implementation

The suggested solution attempts to create a browser extension for phishing detection that is powered by state-of-the-art Machine Learning techniques. The data-set was evaluated using SVM, Random Forest(RF) and Neural Networks methods, and the SVM trained persistent model was sent to the engineering module for phishing detection. The system uses a Support Vector Machine(SVM) trained persistent model to identify phishing websites for classification problem statements because of the margin flexibility and decreased computing complexity provided by SVM.

Three approaches for detecting phishing websites have been devised. The dataset is taken from Kaggle. Various parameters are used like address Bar based features, abnormal based features, HTML and JavaScript based features and Domain based features.

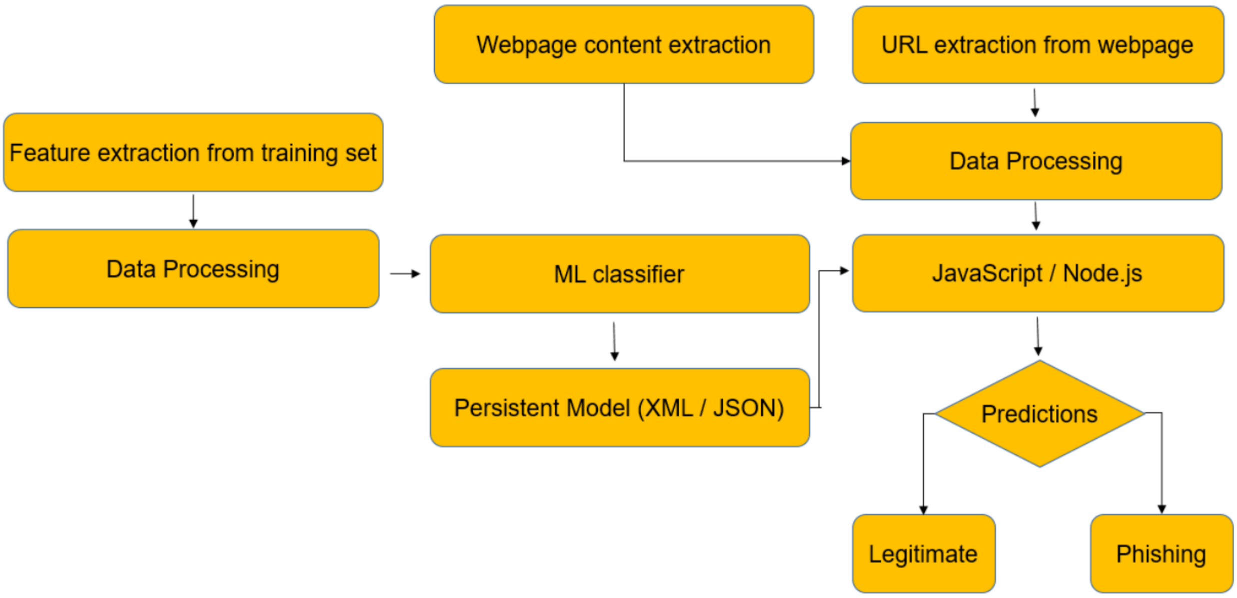

Webpage content extraction
URL extraction from webpage
Feature extraction from training set
Data Processing
Data Processing
ML classifier
JavaScript / Node.js
Persistent Model (XML / JSON)
Predictions
Legitimate
Phishing

Figure 4: Phishing detection in Chrome extension Implementation using machine Learning

The approach combines Python-based training stage implementation with a testing module written in JavaScript. The training aspect of this model was created in Python to make the best use of the available complicated numeric computing tools. The Chrome extension follows to Google standards and is composed largely of three major files: manifest.json, content.js, and display.js. All the meta-data information of the extension for the browser is provided by manifest file that is manifest.js. This file gives Chrome basic information about the extension, such as its name, files, permissions, and relevant scripts. Following the deployment of the extension, the content.js file is loaded on every page in the browser. It operates in a separate unprivileged JavaScript environment with full Document Object Model (DOM) access. In this case, the trained SVM model (weights computed using Python's Machine Learning method) was utilized as a persistent model to categorize webpages. The predicted value for the website is computed using the evaluated feature vector, which is then provided to the predict(data) function. “display.js”, a secondary file, assists the content script with these connections, which are known as message forwarding. If we require access to external extensions or APIs, we must build a channel of interaction between content.js and accessible sections of our extension. Message forwarding enables our extension's various components to communicate with one another. It is, nevertheless, an unprivileged module with direct access only to the DOM components and requires supporting files to communicate with external APIs and modify the browser user experience. The "content.js" script includes several methods for extracting site content and URL features. The details required to detect phishing websites are listed below.

isIPInURL(): Checks if IP address is used in URL.

isLongURL():Checks if length of the URL is beyond 75 characters.

isTinyURL(): Find URLs with fewer than 20 characters.

isAlphaNumericURL(): Examine the URL for the alphanumeric character '@'

isRedirectingURL(): Check to see if the string'//' appears more than once in the URL.

isHypenURL(): Examine the URL for the existence of a '-' next to the domain name.

isMultiDomainURL(): Only top-level domains, country codes, and second-level domains should be used in domain names.

isFaviconDomainUnidentical(): Check to see whether any links on a particular web page are retrieved from other domains.

isIllegalHttpsURL(): Determines the existence of multiple 'HTTPS' characters in the URL string.

isImgFromDifferentDomain(): Determine whether images on a particular web page are retrieved from other domains.

isAnchorFromDifferentDomain(): Determine whether links on a particular website are retrieved from other domains.

isScLnkFromDifferentDomain(): Determine whether scripts from other domains are imported on a particular web page.

isFormActionInvalid(): Form submissions that are incorrect or blank are detected.

isMailToAvailable(): Look for anchor tags with "mailto:" in them.

isStatusBarTampered(): Check to see if "onMouseOver" affects the status bar display.

isIframePresent(): Determine which sites have iframes in their Document Object Model (DOM).

The data is then given to the Support Vector Machine(SVM) classifier after the features have been chosen. If the result is 1, the website is safe, else, the website has been phished.

5 Evaluation

On the phishing dataset, this project compares the performance of all the classifiers given in section 3. We tested these algorithms on 3317 test samples using various performance indicators, and the results are reported and graphed in this part.

5.1 Random Forest(RF) Confusion Matrix

	Predicted Phishing URLs	Predicted Legitimate URLs
Real Phishing URLs	1249	162
Real Legitimate URLs	182	1680

Table 1: The confusion matrix for Random forests

1249 true positives(TP), 162 false positives(FP), 182 false negatives(FN), and 1680 true negatives(TN) were recorded in Table 1.

5.2 Artificial Neural Network(ANN) Confusion Matrix

	Predicted Phishing URLs	Predicted Legitimate URLs
Real Phishing URLs	1205	250
Real Legitimate URLs	170	1692

Table 2: The confusion matrix for ANN

1205 true positives(TP), 250 false positives(FP), 170 false negatives(FN), and 1692 true negatives(TN) were recorded in Table 2.

5.3 Support Vector Machine(SVM) Confusion Matrix

	Predicted Phishing URLs	Predicted Legitimate URLs
Real Phishing URLs	1293	206
Real Legitimate URLs	131	1731

Table 3: The confusion matrix for SVM

1293 true positives(TP), 206 false positives(FP), 131 false negatives(FN), and 1731 true negatives(TN) were recorded in Table 3.

5.4 Performance Matrix Graphs of Classifiers

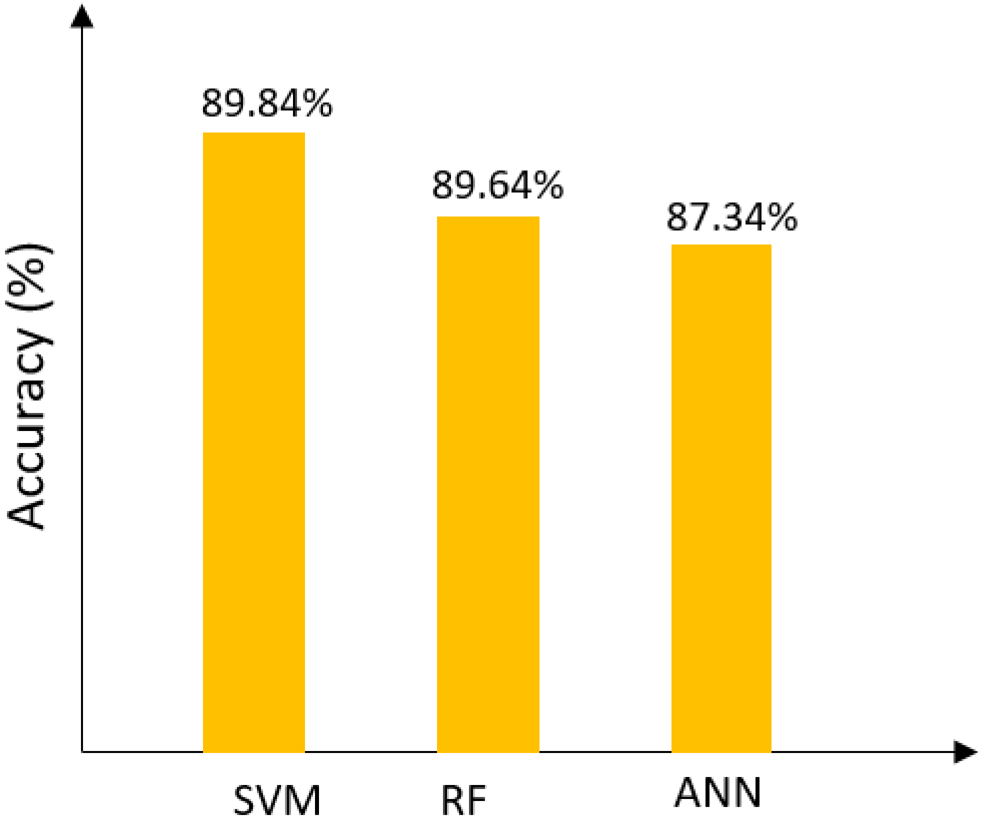


Figure 5: Accuracy of Classifiers

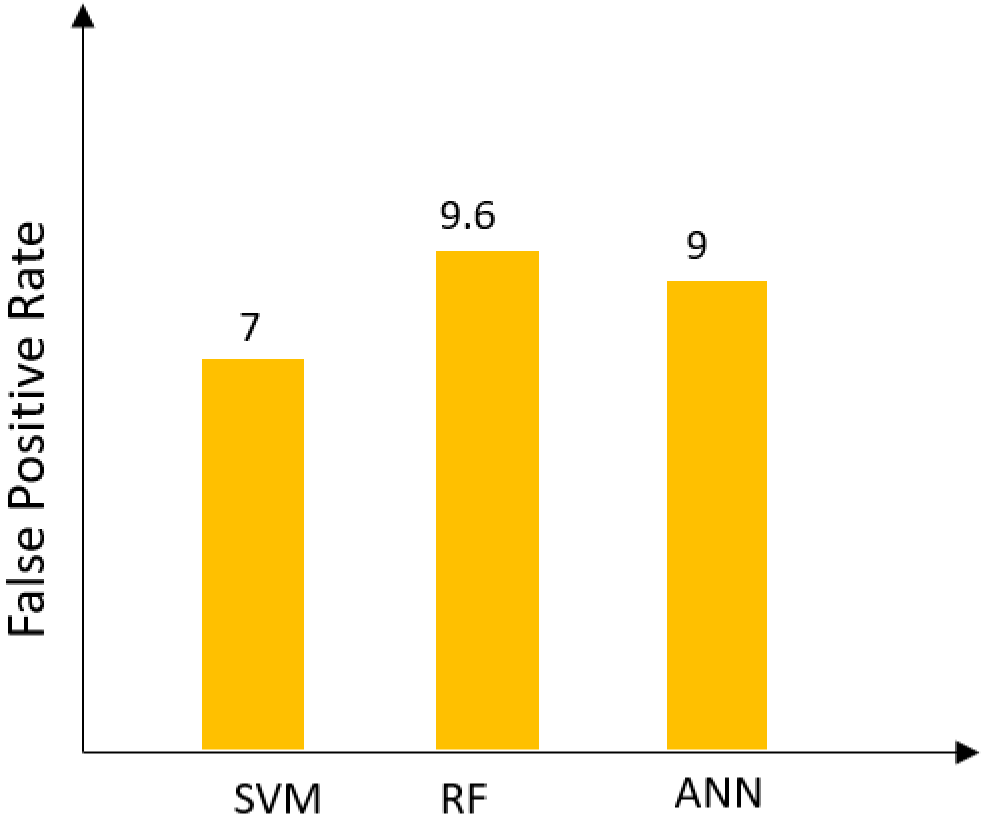


Figure 6: False Positive Rate(FPR) of Classifiers

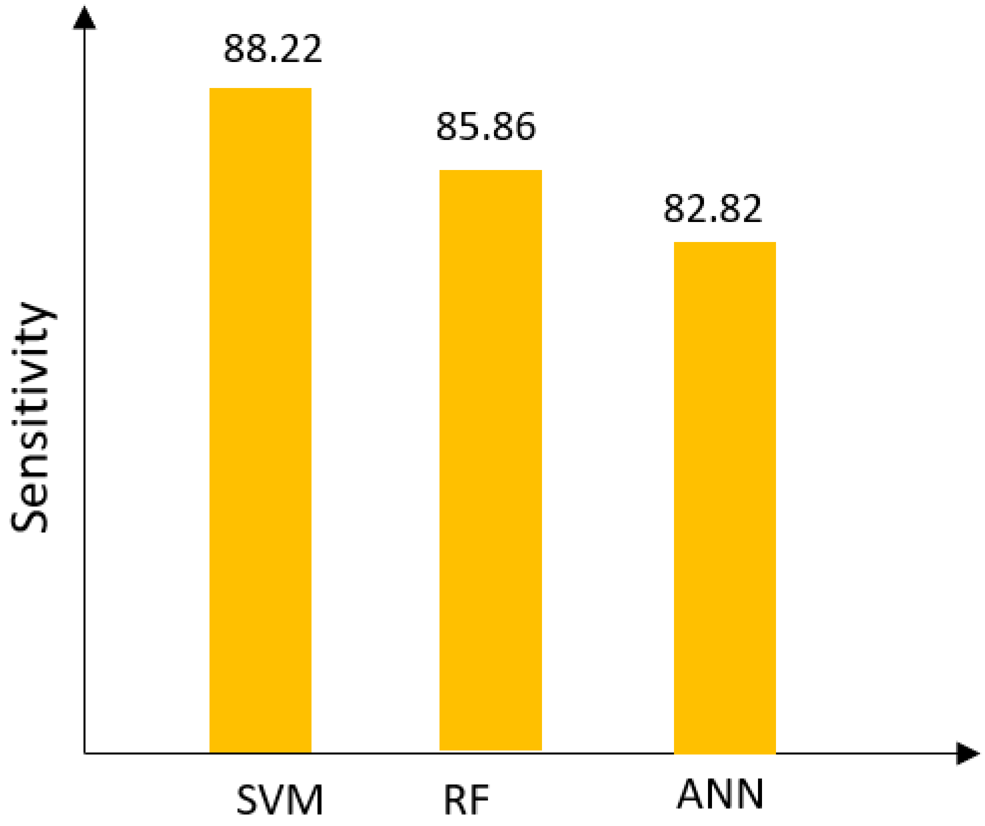


Figure 7: Sensitivity of Classifiers

5.5 Discussion

	Accuracy(%)	Sensitivity(%)	Specificity(%)
SVM	**89.84**	**88.22**	**93**
RF	**89.63**	**85.86**	**90**
ANN	**87.34**	**82.82**	**91**

Table 4: Performance matrix of classifiers

Figure 5 depicts the accuracy of each classifier based on 3317 test samples. According to the graph above, Support Vector Machine(SVM) beats all other algorithms in terms of accuracy in detecting Phishing URLs. Figure 7 also depicts the sensitivity of each classifier. Sensitivity relates to the classifier's ability to recognize phishing URLs properly. As can be shown, SVM has the highest sensitivity of all the classifiers. In phishing detection, however, false positives and false negatives are valued more highly when analysing a classifier's performance which is also known as predictive accuracy. In the actual world, false positives are more harmful than false negatives. Because we don't want consumers to be able to visit the phishing URLs, false positives are taken into account while determining the best classifier. Figure 6 depicts the false positive rates for all classifiers. SVM has the lowest false positive rate among the three. In Table 4, overall comparison or performance matrix of the classifiers are shown. According to the table 4, SVM has the best results in all the evaluation metric we have performed. SVM has the highest accuracy, sensitivity and specificity rates and lowest false positive rates(FPR). As a result, Support Vector Machine(SVM) performs best in distinguishing between legitimate and phishing URLs.

6 Conclusion and Future Work

To conclude, we've seen how phishing poses a significant danger to web security and safety, and how phishing detection is a critical issue domain. We examined several of the classic approaches to phishing detection, namely blacklisting, whitelisting and heuristic assessment methods, as well as their limitations. The best algorithm was then chosen based on its performance, and a Chrome plugin for identifying phishing web sites was created. The plugin makes it simple to implement our phishing detection algorithm to end users. The proposed technique used Support Vector Machine(SVM) with an accuracy of 89.84 percent and a very low false positive rate. The presented system can detect new temporary phishing sites and mitigate the harm caused by phishing assaults. In the future, we want to design the phishing detection system as a scalable web service with online learning and on even larger dataset so that new phishing attack patterns may be readily learnt and to increase the accuracy of our models with improved feature extraction. We would also like to use deep learning methods with backpropagation to attain the utmost accuracy and lowest false positive rates.

References

[1] Basit, A., Zafar, M., Liu, X. et al. A comprehensive survey of AI-enabled phishing attacks detection techniques. Telecommun Syst 76, 139–154 (2021). https://doi.org/10.1007/s11235-020-00733-2

[2] H. Shahriar and M. Zulkernine, "PhishTester: Automatic Testing of Phishing Attacks," 2010 Fourth International Conference on Secure Software Integration and Reliability Improvement, 2010, pp. 198-207, doi: 10.1109/SSIRI.2010.17.

[3] E. Gandotra and D. Gupta, “An Efficient Approach for Phishing Detection using Machine Learning,” Multimedia Security Algorithms for Intelligent Systems, pp. 239–253, 2021.

[4] C. L. Tan, K. L. Chiew, K. S. Wong, and S. N. Sze, “PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder,” Decision Support Systems, 01-Jun-2016. [Online].

[5] Jain, A.K., Gupta, B.B. Towards detection of phishing websites on client-side using machine learning based approach. Telecommun Syst 68, 687–700 (2018).

[6] M. Moghimi and A. Y. Varjani, “New rule-based phishing detection method,” Expert Systems with Applications, 27-Jan-2016. [Online].

[7] R. B. Basnet and T. Doleck, "Towards Developing a Tool to Detect Phishing URLs: A Machine Learning Approach," 2015 IEEE International Conference on Computational Intelligence & Communication Technology, 2015, pp. 220-223, doi: 10.1109/CICT.2015.63.

[8] C. -Y. Wu, C. -C. Kuo and C. -S. Yang, "A Phishing Detection System based on Machine Learning," 2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA), 2019, pp. 28-32, doi: 10.1109/ICEA.2019.8858325.

[9] Rao, R.S., Pais, A.R., “Detecting Phishing Websites using Automation of Human Behavior”| Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security, 01-Apr-2017. [Online].

[10] Xiaoqing GU, Hongyuan WANG, and Tongguang NI, “http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.353.4273&rep=rep1&type=pdf,” Journal of Computational Information Systems, vol. 9, no. 14, pp. 1–8, 2013.

[11] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, “Machine learning based phishing detection from URLs,” Expert Systems with Applications, 18-Sep-2018. [Online].

[12] A. Subasi, E. Molah, F. Almkallawi and T. J. Chaudhery, "Intelligent phishing website detection using random forest classifier," 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), 2017, pp. 1-5, doi: 10.1109/ICECTA.2017.8252051.

[13] Feng, F., Zhou, Q., Shen, Z. et al. The application of a novel neural network in the detection of phishing websites. J Ambient Intell Human Comput (2018).

[14] S. Shukla and P. Sharma, "Detection of Phishing URL using Bayesian Optimized SVM Classifier," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2020, pp. 1385-1389, doi: 10.1109/ICECA49313.2020.9297412.

[15] S. Sindhu, S. P. Patil, A. Sreevalsan, F. Rahman and M. S. A. N., "Phishing Detection using Random Forest, SVM and Neural Network with Backpropagation," 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), 2020, pp. 391-394, doi: 10.1109/ICSTCEE49637.2020.9277256.

[16] S. Sindhu, S. P. Patil, A. Sreevalsan, F. Rahman and M. S. A. N., "Phishing Detection using Random Forest, SVM and Neural Network with Backpropagation," 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), 2020, pp. 391-394, doi: 10.1109/ICSTCEE49637.2020.9277256.

[17] Chand, E. (2020, February 28). Phishing website Detector.

[18] Y. Guo, Y. Zhou, X. Hu and W. Cheng, "Research on Recommendation of Insurance Products Based on Random Forest," 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), 2019, pp. 308-311, doi: 10.1109/MLBDBI48998.2019.00069.

[19] IC3 Annual Report 2018 https://pdf.ic3.gov/2018_IC3Report.pdf

[20] A. P. E. Rosiello, E. Kirda, 2. Kruegel and F. Ferrandi, "A layout-similarity-based approach for detecting phishing pages," 2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007, 2007, pp. 454-463, doi: 10.1109/SECCOM.2007.4550367.

[21] E. Kirda and C. Kruegel, "Protecting users against phishing attacks with AntiPhish," 29th Annual International Computer Software and Applications Conference (COMPSAC'05), 2005, pp. 517-524 Vol. 2, doi: 10.1109/COMPSAC.2005.126.

[22] D. Irani, S. Webb, J. Giffin and C. Pu, "Evolutionary study of phishing," 2008 eCrime Researchers Summit, 2008, pp. 1-10, doi: 10.1109/ECRIME.2008.4696967.

[23] M. Sharifi and S. H. Siadati, "A phishing sites blacklist generator," 2008 IEEE/ACS International Conference on Computer Systems and Applications, 2008, pp. 840-843, doi: 10.1109/AICCSA.2008.4493625.

[24] Y. Wang, R. Agrawal and B. Choi, "Light Weight Anti-Phishing with User Whitelisting in a Web Browser," 2008 IEEE Region 5 Conference, 2008, pp. 1-4, doi: 10.1109/TPSD.2008.4562720.

[25] J. Kang and D. Lee, "Advanced White List Approach for Preventing Access to Phishing Sites," 2007 International Conference on Convergence Information Technology (ICCIT 2007), 2007, pp. 491-496, doi: 10.1109/ICCIT.2007.50.

[26] says:, S. A., says:, V., says:, S., says:, M. E., says:, D., says:, B. K., … Says:, U. (2017, September 13). Basic evaluation measures from the confusion matrix.

[27] Y. Yang, J. Wang and Y. Yang, "Improving SVM classifier with prior knowledge in microcalcification detection1," 2012 19th IEEE International Conference on Image Processing, 2012, pp. 2837-2840, doi: 10.1109/ICIP.2012.6467490.

[28] B. Sanjaa and E. Chuluun, "Malware detection using linear SVM," Ifost, 2013, pp. 136-138, doi: 10.1109/IFOST.2013.6616872.

[29] "Phishing Website Dataset," *UCI Machine Learning Repository: Phishing Websites Data Set*. [Online].