Volume: 11 Issue: XII Month of publication: December 2023
DOI: https://doi.org/10.22214/ijraset.2023.57540

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue XII Dec 2023- Available at www.ijraset.com

Detecting Phishing Website Using Machine Learning

Abhishek Tripathi[1], Anurag Rao[2], Hiransh Singh[3], Prof. Dr. Shashi Kant Singh[4]
Department of Computer Science and Engineering, Galgotias College of Engineering and Technology, Gautam Buddha Nagar, India

Abstract: The Internet has grown to be an essential aspect of our lives, but it has also given rise to instances of harmful activity, such as phishing, that may be carried out anonymously. Phishers attempt to trick their victims by using social engineering techniques or building dummy websites in order to get personal data from people and businesses, including usernames, account IDs, and passwords. Despite the fact that several techniques have been put out to identify phishing websites, scammers have developed ways to evade detection. Machine Learning is one of the best techniques for identifying these harmful actions. This is due to the fact that machine learning techniques can recognise some common traits shared by the majority of phishing assaults. In order to anticipate phishing websites, we compared the outcomes of many machine learning techniques in this article.
Keywords: Phishing, Categorization, Cybercrime, and Machine-learning

I. INTRODUCTION

Phishing is a type of cybercrime that involves building a fake website that seems like a genuine website in an attempt to trick people into providing sensitive or vital information. Phishing attacks include a range of strategies, including social engineering, website forgery, covert redirection, filter evasion, and link manipulation. Creating a spoof website that mimics a genuine website is the most popular method. The U.S. Federal Bureau of Investigations' Internet Crime Complaint Centre (IC3) highlighted these kinds of assaults as top issues in its most recent 2018 Internet Crime Report. The FBI's IC3 figures for 2018 demonstrated that online theft, fraud, and abuse are still common and caused an astounding $2.7 billion in damages to the financial system

With losses exceeding $1.2 billion, the IC3 received 20,373 complaints about email account compromise (EAC) and corporate email compromise (BEC) in that year [1]. The report notes that the number of these sophisticated attacks has increased in recent years. The Anti-Phishing Working Group (APWG) highlights that phishing attacks have increased in recent years. Figure 1 shows the total number of phishing sites discovered by APWG in the first quarter of 2020 and the last quarter of 2019. This figure shows a gradual increase from 162,155 in the fourth quarter of 2019 to 165,772 in the first quarter of 2020. Phishing is causing serious damage to many organizations and the global economy, and in Q4 2019, his APWG member His OpSec Security found that SaaS . Webmail sites also continued to be the most common targets for phishing attacks. The phisher continues to run her BEC to collect credentials from these targets and access her SaaS account at the company [2]. There are many ways to filter out phishing websites.

Each of these methods can be applied to different phases of the attack flow, including network-level protection, authentication, client-side tools, user training, and server-side filters and classifiers. Although each type of phishing attack has some unique characteristics, most of these attacks share some commonalities and patterns.

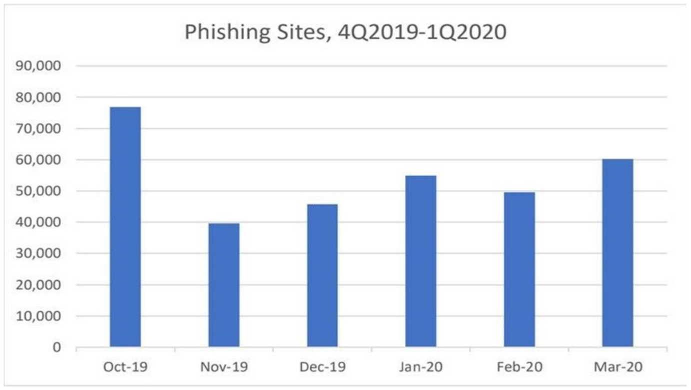


Fig. 1. Total number of phishing websites detected by APWG

Machine learning techniques have proven to be powerful tools for identifying patterns in data, so these techniques You can now detect some of the characteristics of phishing recognizing phishing websites. This article provides a comparative and analytical evaluation of various machine learning techniques for detecting phishing websites. The machine learning techniques we will consider are Logistic Regression, Decision Trees, Random Forests, Ada-Boost, Support Vector Machines, ANNs, Artificial Neural Networks, Gradient Boosting, and XGBoost. The remainder of the paper is structured as follows: Section II lists some popular phishing approaches, and Section III discusses various phishing techniques and ways to thwart phishing attacks. A summary of the various machine learning techniques for phishing detection is given in Section IV. Section V describes the features of our dataset. Sections VI and VII present the evaluation results of the proposed machine learning method, and finally, Section VIII draws conclusions and discusses future work.

II. PHISHING TECHNIQUES

This section describes common phishing techniques used by criminals to trick people.

A. Link manipulation

Link Manipulation Phishing is primarily about links. There are some clever ways to make a URL look like a legitimate URL. One method is to display the malicious URL as a named hyperlink on your website. Another method is to use a misspelled URL that looks like a legitimate URL, such as ghoogle.com. A variant of typo squatting that is much more difficult to detect than the link manipulation methods described above is so-called IDN spoofing. Attackers use non-English characters (such as Cyrillic) that look exactly like English characters. One "c" or "a". ” rather than its English counterpart [3]. Filter Bypass Phishers display website content as images or use Adobe Flash, making them difficult to detect by some phishing detection methods. Avoiding this type of attack requires the use of optical character recognition [4].

B. Filter evasion

Filter Bypass Phishers display website content as images or use Adobe Flash, making them difficult to detect by some phishing detection methods. Avoiding this type of attack requires the use of optical character recognition [4].

C. Website forgery

Website forgery This type of attack involves phishing a legitimate website by manipulating the target website's JavaScript code. This type of attack, also known as cross-site scripting, is very difficult to detect because the victim is using her legitimate website.

D. Covert redirect

Covert Redirect This attack targets websites that use OAuth 2. 0 and OpenID protocols. When attempting to grant token access to her legitimate website, the user hands over his token to the malicious service. However, this method has not received much attention due to its low information value [5].

E. Social engineering

Social Engineering This type of phishing is carried out through social interactions. It uses psychological tricks to trick users into revealing security information. This type of attack occurs in several stages. First, phishers examine potential vulnerabilities in the target they want to attack. The phishers then try to gain the trust of their targets and ultimately create a situation where the targets divulge sensitive information. Social engineering phishing techniques include decoys, scareware, pretexting, and spear phishing [6].

III. PHISHING DETECTION APPROACHES: AN OVERVIEW

Various methods have been proposed to mitigate phishing attacks at each level of the attack flow. Some of these methods require user training to prepare for future attacks, while others work automatically and alert the user. These methods can be listed as follows:

A. User Training

Educating and warning users and company employees about phishing attacks can help prevent them.Several methods have been proposed for user training. Many studies have concluded that interactive instructions are the most effective approach for helping users distinguish between phishing and legitimate websites [7][8].

Although user training is an effective method however humans errors still exist and people are prone to forget their training. Training also requires a significant amount of time and it is not much appreciated by non-technical users [9].

B. Software Detection

Even while user training can stop some phishing attacks, there are hundreds of websites that we are exposed to every day, so using our training on each one is a laborious and occasionally impractical effort. Another alternative to detecting phishing websites is to use software. This software can analyze multiple factors such as website content, email messages, URLs, and many other features before making a final decision, which is more reliable than humans.

Several software techniques have been proposed for phishing detection, categorized as follows:

1) *List-base approach*: One of the widely used methods for phishing detection is using blacklist-based anti-phishing methods which are integrated into web browsers. These methods use two types of lists: whitelists, which contain the names of valid websites, and blacklists, which contain a list of malicious websites. Blacklists are typically created through user feedback or third-party reports generated using different phishing detection schemes. Some studies have shown that blacklist-based anti-phishing approaches can detect 90 percent of the malicious website at the time of initial check [10].
2) *Visual similarity-base approach*: One of the main reasons that people are tricked into believing that they are using a legitimate website but in reality, they are filling a form A characteristic of malicious websites is that the phishing website looks exactly like the legitimate website being attacked. Some methods exploit visual similarities to identify phishing websites by analyzing the text content, text format, HTML, CSS, and images of web pages [11] [12]. Chen et al [13] also proposed discriminative keypoint features considering phishing detection as an image matching problem. Visual similarity-based approaches have limitations. For example, the method that uses website content cannot detect websites that use images instead of text. Methods using image matching methods are very time-consuming and difficult to collect sufficient data [14].
3) *Heuristics and machine learning based*: Machine learning methods have proved to be a powerful tool to classify malicious activities or artifacts like spam emails or phishing websites. Most of these methods require training data. Fortunately, there are many examples of phishing websites for training machine learning models. Some machine learning methods use vision techniques by analyzing snapshots of websites [15]. Others may use website content and functionality for phishing detection. Several machine learning techniques are used to detect phishing websites. Some of them are Logistic Regression, Decision Trees, Random Forests, Ada Boost, SVM, KNN, Neural Networks, Gradient Boosting, and XGBoost, which are discussed in the next section. In a recent study [16] on phishing, the authors emphasized that when some new solutions were proposed to overcome various phishing attacks, attackers evolve their method to bypass the newly proposed phishing method. Therefore, we strongly recommend the use of hybrid models and machine learning-based techniques.

In this paper, we will detect phishing websites using classifiers based on machine learning..

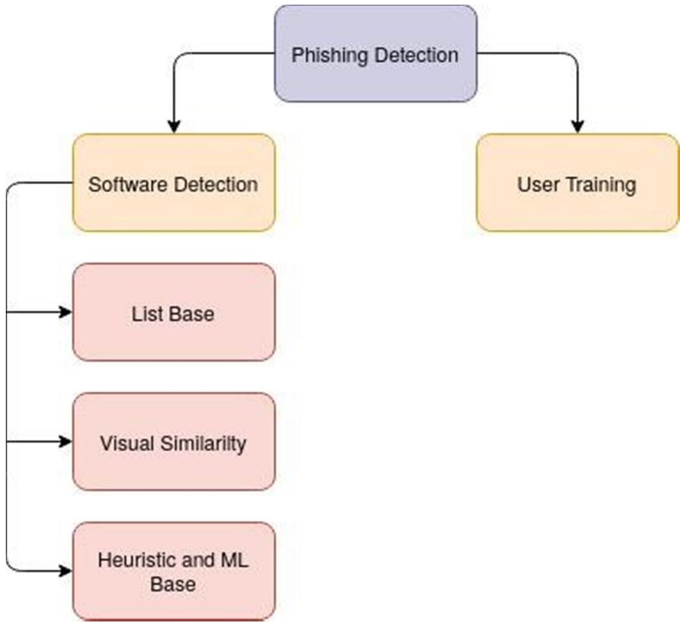


Fig. 2. An Overview of phishing detection approaches

IV. MACHINE LEARNING APPROACH

Machine learning provides a simplified and efficient method for data analysis. It has indicated promising outcomes in realtime classification problems recently. The key advantage of machine learning is the ability to create flexible models for specific tasks like phishing detection. Since phishing is a classification problem, machine learning models can be used as powerful tools. Machine learning models could adapt to changes quickly to identify patterns of fraudulent transactions that help to develop a learning-based identification system. Most of the machine learning models discussed here are classified as supervised machine learning.Here, the algorithm attempts to learn a function that maps inputs to outputs based on example input-output pairs. It derives a function from labeled training data consisting of a set of training samples. We will introduce the machine learning methods used in our research.

A. Logistic Regression

Logistic Regression Logistic regression is a classification algorithm that assigns observations to a discrete set of classes.
Unlike linear regression, which outputs continuous numbers, logistic regression uses a logistic sigmoid function to transform the output and return probability values that can be assigned to two or more discrete classes. Logistic regression works well when the relationships in the data are close to linear.However, complex nonlinear relationships between variables degrade performance. Additionally, further statistical assumptions are required before using other techniques.

B. K Nearest Neighbors

One of the most straightforward non-parametric techniques for regression and classification issues in machine learning is K-Nearest Neighbors (KNN). ANN requires no assumptions about the underlying data distribution. KNN algorithm uses feature similarity to predict the values of new datapoints which means that the new data point will be assigned a value based on how closely it matches the points in the training set. Similarity between datasets can be measured in various ways. Once neighborhoods are found, summary predictions can be made by returning or averaging the most common results. Therefore, ANN can be used for classification and regression problems. Nothing says a model other than storing the entire training data set.

C. Support Vector Machine

Support Vector Machine among the most widely used classifiers is the Support Vector Machine (SVM). SVM uses the maximum distance between two classes to find the point that is closest to them. This technique is a supervised learning model used for linear and nonlinear classification.
Nonlinear classification is performed using a kernel function to map the input to a high-dimensional feature space. SVMs have some weaknesses despite being extremely powerful and frequently used in classification. They need high calculations to train data. Also, they are sensitive to noisy data and are therefore prone to overfitting. The four common kernel functions at the SVM are linear, RBF (radial basis function), sigmoid, and polynomial, which is listed in TableI. Each kernel function has particular parameters that must be optimized to obtain the best result.

TABLE I
Four Common Kernels [17]

Kernel Type	Formula	Parameter
Linear	$n \quad i\ K(xn, xi) = (xnn, xi^{)}\ i\ 2+ {}_{C})$	C,γ
RBF	$K(x\ ,\ x\ ) = exp(-\gamma \lVert x - x \rVert$	C,γ,r
Sigmoid	$K(x_n, x_i) = tanh(\gamma(x_n,x_i) + r)$	C,γ
Polynomial	$K(x_n, x_i) = (\gamma(x_n, x_i) + r)^d$	C,γ,r,d

D. Decision Tree

Support Vector Machine Support Vector Machine (SVM) is one of the most popular classifiers. The idea behind SVM is to find the closest point between two classes using the maximum distance between them. This technique is a supervised learning model used for linear and nonlinear classification.
It can interpret the interaction between predictors. It can also be interpreted very well because of its binary structure. However, the decision tree has various drawbacks that tend to overuse data. Additionaly, it is difficult to update the decision tree with new samples.

E. Random Forest

As the name suggests Random Forest contains many individual decision trees that work as a group to determine the output In a random forest, every tree indicates the class it predicts, and the outcome is the most predicted class among the trees' choices. The trees shield one another from individual mistakes, which is why Random Forest produced such an incredible result. Some trees may predict the wrong answer, but many others will modify the final prediction, allowing the trees as a group to move in the right direction. Because Random Forest uses only a subset of all training samples, it achieves overfitting reduction by combining many weak learners with below-average fitness. Random forests can handle large numbers of variables in a data set. It also provides an unbiased estimate of the generalization error during the forest establishment process. You can also easily estimate lost data. The main drawback of random forests is that the forest construction process is random and therefore not reproducible. Furthermore, interpreting the final model and subsequent results is difficult because many independent decision trees are involved.[18]

F. Ada-Boost

It also provides an unbiased estimate of the generalization error during the forest establishment process. You can also easily estimate lost data. The main drawback of random forests is that the forest construction process is random and therefore not reproducible. Additionally, the final model and subsequent results are difficult to interpret because many independent models are involved. In some ways, Ada-Boost is similar to a random forest. Ada-Boost classification is similar to random forests and groups weak classification models into strong classifiers. A single model cannot adequately classify objects.
However, combining multiple classifiers by selecting a set of samples in each iteration and assigning sufficient weights to the final vote can be beneficial to the overall classification. The trees are built one at a time as weak learners, correcting mispredicted samples by assigning larger weights after each round of prediction. The model learns from previous mistakes. The final prediction is a weighted majority vote (or weighted median for regression problems). In other words, the Ada-Boost algorithm selects and iterates training sets based on the accuracy of previous training. The weights of each classifier trained in each iteration depend on the accuracy of the previous classifier [19].
A single model may poorly categorize objects.

G. Gradeint Boosting

Gradient Boosting trains many models incrementally and sequentially. Gradient Boosting Gradient boosting trains many models incrementally and sequentially. The main difference between Ada-Boost and gradient boosting algorithms is how the algorithm identifies shortcomings in weak learners such as decision trees. The Ada Boost model uses high weight data points to identify defects, while Gradient Boosting uses the gradient of the loss function to perform the same method. A loss function is a measure of how well the model coefficients fit the underlying data. The logical understanding of loss functions depends on what you want to optimize.

H. XGBoost

The XGBoost runs more than ten times faster than popular solutions on a single machine and scales to billions of examples in distributed or memory-limited settings. XGBoost's scalability can be attributed to multiple significant algorithmic improvements. XGBoost is an improved and customized version of gradient boosting to provide better performance and speed. The most important factor for XGBoost's success is scalability in all scenarios. XGBoost runs more than 10 times faster than current solutions on a single computer and can scale to billions of samples in distributed and memory-constrained environments. XGBoost's scalability is due to several important algorithmic optimizations.These innovations include new tree learning algorithms for processing sparse data. A theory-based weighted quantile sketching method allows handling of instance weights in approximate tree learning. Parallel distributed computing accelerates learning and enables faster model exploration. More importantly, XGBoost leverages out-of-core computation, allowing data scientists to process hundreds of millions of samples on their desktop. Finally, and even more interestingly, these techniques can be combined to create an end-to-end system that can scale to even larger amounts of data using minimal cluster resources.

I. Artificial Neural Networks

These models are multilayered, each layer containing several processing units called neurons Artificial neural networks (ANNS) are learning models based largely on biological neural networks. These models are multilayered, with each layer containing multiple processing units called neurons.

Each neuron receives inputs from neighboring layers and uses its weights and a nonlinear function called an activation function to compute an output. In a feedforward neural network like in3, data flows from the first layer to the last layer. In feed-forward neural networks like in3, data flows from the first layer to the last layer. Different layers can perform different transformations for their inputs. The neuron weights are randomly set at the beginning of training and gradually adjusted using gradient descent to approach the optimal solution. The power of neural networks is due to the non-linearity of hidden nodes.The performance of neural networks depends on the nonlinearity of hidden nodes. Therefore, it is very important to introduce nonlinearity into networks so that complex functions can be learned [22].

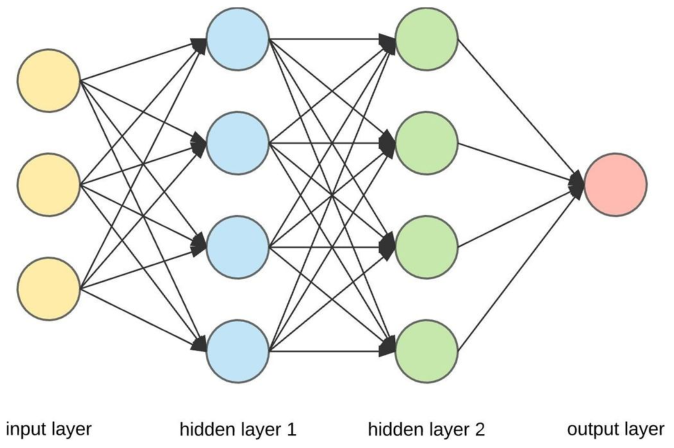


Fig. 3. Artificial Neural Network

V. DATA SET DESCRIPTION

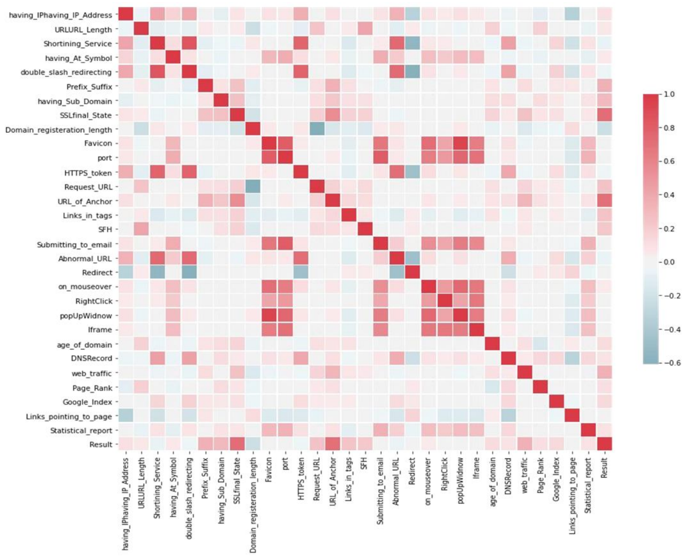
One of the biggest challenges in our research was the lack of phishing datasets. Although many academic papers have been published on phishing detection, the datasets used in the research are not provided. Additionally, another factor complicating the search for the desired dataset is the lack of a standard set of features for recording the characteristics of phishing websites. The datasets we used in our study have been well explored and benchmarked by several researchers.

Fortunately, the Wiki that accompanies the dataset includes a data description document that describes the data generation strategy of the dataset author [23].

To update the dataset with new phishing websites, we also implemented code to extract new phishing website features provided by the PhishTank website. The dataset contains about 11,000 sample websites, we used 10% of samples in the testing phase. Each website is marked either legitimate or phishing. The features of our dataset are as follows:

1) *Having IP Address:* If an IP address is used instead of the domain name in the URL, such as http://217.102.24.235/sample.html.
2) *URL Length:* Phishers can use a long URL to hide the doubtful part in the address bar.
3) *Shortening Service:* Links to the webpage that has a long URL. For example, the URLhttp://sharif.hud.ac.uk/ can be shortened to bit.ly/1sSEGTB.
4) *Having @ Symbol:* Using the @ symbol in the URL leads the browser to ignore everything preceding the @ symbol and the real address often follows the @ symbol
5) *Double Slash Redirection:* The existence of // within the URL which means that the user will be redirected to another website
6) *Prefix Suffix:* Phishers tend to add prefixes or suffixes to domain names separated by (-) to trick users into thinking they are dealing with a legitimate website. For examplehttp://www.Confirme-paypal.com.
7) *Having Sub Domain:* Having subdomain in URL.
8) *SSL State:* Shows that website use SSL
9) *Domain Registration Length:* Based on the fact that a phishing website lives for a short period
10) *Favicon:* A favicon is a graphic image (symbol) associated with a particular web page. If the favicon is loaded from a domain other than that shown in the address bar, thenthe webpage is likely to be considered a Phishing attempt.
11) *Using Non-Standard Port:* To control intrusions, it is much better to merely open ports that you need. By default, some firewalls, proxies, and network address translation (NAT) servers block all or most ports and open only selected ports.
12) *HTTPS Token:* Having deceiving https token in URL.For example,http://https-www-mellat-phish.ir

13) *Request URL:* Request URL examines whether the external objects contained within a webpage such as images, videos, and sounds are loaded from another domain.
14) *URL of Anchor:* An anchor is an element defined by the < *a* >tag. This feature is treated exactly as Request URL.
15) *Links in Tags:* Meta tags are frequently used by trustworthy websites to provide metadata about the HTML document, and Script tags are frequently used to create client-side scripts.
16) *Server Form Handler:* If the webpage's domain name differs from the domain name in SFHs.
17) *Submitting Information To E-mail:* A phisher might redirect the users information to his email.
18) *Abnormal URL:* It is extracted from the WHOIS database. For legitimate websites, the ID is usually part of its URL.
19) *Website Redirect Count:* If the redirection is more than four-time
20) *Status Bar Customization:* Use JavaScript to show a fake URL in the status bar to users
21) *Disabling Right Click:* It is treated exactly as Using onMouseOver to hide the Link
22) *Using Pop-up Window:* Showing having popo-up windows on the webpage.
23) *IFrame:* IFrame is an HTML tag used to display an additional webpage into one that is currently shown.
24) *Age of Domain:* If the age of the domain is less than a month.
25) *DNS Record:* Having the DNS record
26) *Web Traffic:* This feature measures the popularity of TABLE II the website by determining the number of visitors. DESCRIPTION OFDATASET 27)
27) *Page Rank:* Page rank is a value ranging from 0 to 1. PageRank aims to measure how important a website is on the internet.
28) *Google Index:* This feature examines whether a website is in Googles index or not.
29) *Links Pointing To Page:* The number of links pointing to the web page.
30) *Statistical Report:* If the IP among the top phishing IPs ?



$$acc = \frac{N_{L\to L} + N_{P\to P}}{N_{L\to L} + N_{L\to P} + N_{P\to L} + N_{P\to P}}$$

$$r = \frac{N_{P\to P}}{N_{P\to L} + N_{P\to P}}$$

$$p = \frac{N_{P\to P}}{N_{L\to P} + N_{P\to P}}$$

Fig. 4. Correlation of features in dataset

VI. EVALUATION METRICS

To evaluate the performance of phishing classification, use the classifier's accuracy (acc), recall (r), precision (p), F1 score, testing time, and training time. Recall measures the percentage of phishing websites that the model manages to detect (models effectiveness). Precision measures the degree to which the phishing detected websites are indeed phishing (models safety). The F1 score is a weighted harmonic mean of precision and recall. Let $N_{L\rightarrow L}$ be the number of legitimate websites classified as legitimate, $N_{L\rightarrow P}$ be the number of legitimate websites misclassified as phishing, $N_{P\rightarrow L}$ be the number of phishing misclassified as legitimate and $N_{P\rightarrow P}$ be the number of phishing websites classified as phishing. Thus the following equations hold

F1 = P1+Prr

$$F1 = \frac{2pr}{p+r} \qquad (4)$$

features	mean	std
Having IP Address	0.3137	0.9495
URL Length	-0.6331	0.7660
Shortening Service	0.7387	0.6739
Having @ Symbol	0.7005	0.7135
Double Slash Redirecting	0.7414	0.6710
Prefix Suffix	-0.7349	0.6781
Having Sub Domain	0.0639	0.8175
SSL Final State	0.2509	0.9118
Domain Reg Length	-0.3367	0.9416
Favicon	0.6285	0.7777
Port	0.7282	0.6853
HTTPS Token	0.6750	0.7377
Request URL	0.1867	0.9824
URL of Anchor	-0.0765	0.7151
Links in Tags	-0.1181	0.7639
SFH	-0.5957	0.7591
Submitting To Email	0.6356	0.7720
Abnormal URL	0.7052	0.7089
Website Redirect Count	0.1156	0.3198
On Mouse over	0.7620	0.6474
RightClick	0.9138	0.4059
PopUpWidnow	0.6133	0.7898
IFrame	0.8169	0.5767
Age of Domain	0.0612	0.9981
DNS Record	0.3771	0.9262
Web Traffic	0.2872	0.8277
Page Rank	-0.4836	0.8752
Google Index	0.7215	0.6923
Links Pointing to Page	0.3440	0.5699
Statistical Report	0.7195	0.6944
Result	0.1138	0.9935

VII. EXPERIMENTAL RESULTS

In our research, we assessed the model's performance using 10-fold cross-validation. The data set was split up into ten smaller samples. The remaining sample is utilised to train models, while a subsample is used for testing data. We must employ a binary classification model as phishing detection is a classification problem. We define "-1" as a phishing sample and "1" as a valid sample. We employed a range of machine learning models in our study, including SVM, Gradient Boosting, Ada booster, Random Forest, KNN, neural networks, and logistic regression, to detect phishing websites. Evaluate the accuracy, recall, precision, F1 score, training time, and testing time of these models. To get the best results, we experimented with various feature selection and hyperparameter tuning strategies. Table II presents a comparison.In our research, we assessed the model's performance using 10-fold cross-validation. The data set was split up into ten smaller samples. The remaining sample is utilised to train models, while a subsample is used for testing data. We must employ a binary classification model as phishing detection is a classification problem. We define "-1" as a phishing sample and "1" as a valid sample. We employed a range of machine learning models in our study, including SVM, Gradient Boosting, Ada booster, Random Forest, KNN, neural networks, and logistic regression, to detect phishing websites.We assess these models' precision, recall, accuracy, F1 score, training time, and testing time. To get the best results, we experimented with various feature selection and hyperparameter tuning strategies. Table II presents a comparison.

To determine the best support vector machine performance, we tested four different kernels:

1) Linear kernel
2) Polynomial kernel
3) Sigmoid kernel
4) RBF kernel

TABLE III CLASSIFICATION RESULTS
FOR DIFFERENT METHODS

classifier	train time (s)	test time(s)	accuracy	recall	precision	F1 score
logistic regression	0.080971	0.006414	0.926550	0.943968	0.925700	0.934704
decision tree	0.021452	0.003737	0.965988	0.971414	0.967681	0.969531
random forest	0.436126	0.021941	0.972682	0.981484	0.969852	0.975622
ada booster	0.336519	0.016766	0.936953	0.954362	0.933943	0.944032
KNN	0.112972	0.353562	0.952780	0.962968	0.952783	0.957827
neural network	9.088517	0.006925	0.969879	0.978723	0.967605	0.973112
SVM linear	1.647538	0.053979	0.927726	0.945592	0.926268	0.935779
SVM poly	1.048257	0.074207	0.949254	0.968816	0.941779	0.955083
SVM rbf	1.341540	0.103329	0.952149	0.968815	0.946580	0.957543
SVM sigmoid	1.344607	0.109696	0.827498	0.846515	0.844311	0.845305
gradient boosting	0.891888	0.005298	0.948621	0.962481	0.946234	0.954260
XGBoost	0.506072	0.006237	0.983235	0.981047	0.987235	0.976802

In this study, we evaluated the model's performance by the use of 10-fold cross-validation. Ten smaller samples were created from the original data set.

A subsample is utilised for testing data, while the remaining sample is used to train models. Since phishing detection is a classification problem, we have to use a binary classification model. A sample marked with "-1" has been subjected to phishing, while a sample marked with "1" is authentic.

To identify phishing websites, we used a variety of machine learning models in our study, such as SVM, Gradient Boosting, Ada booster, Random Forest, KNN, neural networks, and logistic regression.We evaluate the precision, recall, accuracy, F1 score, testing time, and training time of these models. We experimented with several feature selection and hyperparameter tweaking techniques to obtain the best results. Table II displays a comparsion.

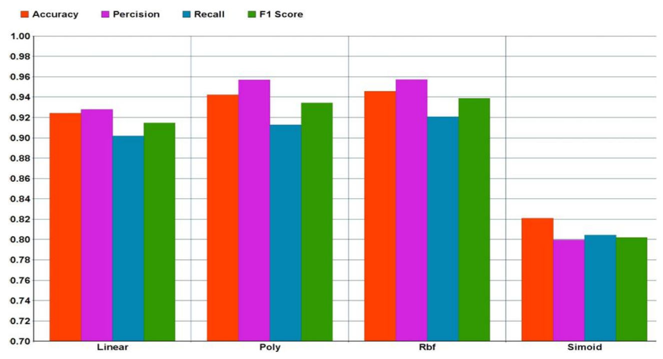
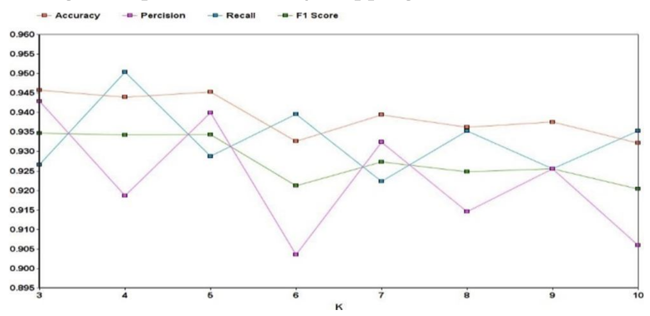


Fig. 5. Performance of SVM classfier with various kernel

We discovered that Random Forest is incredibly accurate, somewhat resilient to noise and anomalies, quick to execute, easy to comprehend, and capable of implicit feature selection. Random Forest's primary benefit over AdaBoost is its immunity to noise. The Central Limit Theorem states that Random Forest lowers variance by growing the tree count. But the biggest challenge we encountered when putting our model into practise using Random Forests was the large number of hyperparameters that needed to be adjusted in order to achieve optimal performance. Furthermore, not all data sets can benefit from Random Forest's introduction of randomization into the training and testing data. We discovered that setting k to 5 yields the greatest results when it comes to KNN categorization. There isn't a single best value for k in KNN classification that works with every type of dataset. The KNN result, which is displayed in Figure 6, indicates that when there are few neighbours, noise will have a greater influence on the outcome. In addition, a large number of neighbours increases the computational cost of obtaining the result. Our findings also indicate that the most flexible fit has few neighbours, resulting in low bias but high variance, and that a large number of neighbours results in a smoother decision boundary, resulting in reduced variance but greater bias. The primary benefits of XGBoost are its quick speed in comparison to other algorithms like ANN and SVM, as well as its effective variance reduction provided by its regularisation parameter. But in addition to using the regularisation parameter, this method also makes use of subsamples and a learning rate from features such as random forests, which improves its capacity to generalise even more. Compared to AdaBoost and Random Forests, XGBoost is more challenging to comprehend, visualise, and tune. To improve performance, a wide range of hyperparameters may be adjusted.When both speed and great accuracy are critical, XGBoost is a very intriguing approach. However, additional resources are needed for model training since model tweaking requires more user skill and effort to do. This algorithm uses subsamples from random forests and a learning rate derived from the regularization parameter, which further improves the algorithm's ability to generalize. However, compared to AdaBoost and Random Forest, XGBoost is more difficult to understand, visualize, and customize. There is a multitude of hyperparameters that can be tuned to increase performance.XGBoost is a particularly interesting algorithm when speed as well as high accuracies are of the essence. Still, model optimization requires more time and user expertise to achieve meaningful results, and model training requires more resources. The training time of a neural network has noticeably longer than that of other machine learning models, as would be expected. The F1 score of XGBoost was marginally higher than that of neural networks. This is because we have a minimal amount of training data. The neural network model, in contrast to XGBoost , is also unable to provide an explaination for why it identified a website as phishing. Explainability will make it easier for us to identify important qualities. Figure 7 illustrates the neural network’s performance with varying numbers of hidden layers; 30 hidden layers yields the best results. Adam optimizer and rely activation function n are used in the neural network implementation. Our model was trained using 500 epochs with early stopping.



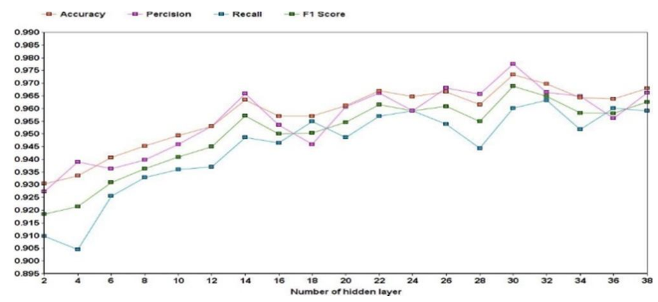


Fig. 7. Neural Network with different depth

VIII. CONCLUSION AND FUTURE WORK

In this study, we implemented and evaluated 12 classifiers on a phishing website dataset consisting of 6157 legitimate websites and 4898 phishing websites. The classifiers investigated are Logistic Regression, Decision Trees, Support Vector Machines, Ada Boost, Random Forests, Neural Networks, KNN, Gradient Boosting, and XGBoost. According to our result in TableIII, we get very good performance in ensembling classifiers namely, Random Forest, XGBoost both on computation duration and accuracy. The main idea behind ensemble algorithms is to combine several weak learners into There are certain advantages and disadvantages inherent to the AdaBoost algorithm.

AdaBoost is relatively robust against overfitting on low-noise datasets. AdaBoost has few hyperparameters that need to be tuned to improve model performance. Moreover, this algorithm is easy to understand and visualize. However, AdaBoost's performance on noisy data is debatable, with some claiming that it generalizes well and others claiming that it takes too long to learn extreme cases, distorting the results, and that Some argue that certain data leads to poor performance. Compared to random forests and XGBoost, Moreover, AdaBoost is not optimized for speed, therefore being significantly slower than K. Krombholz, H. Hobel, M. Huber, and E. Weippl, “Advanced social engineering attacks,”.

XGBoost.: Note that there is no guarantee that a combination of multiple classifiers will always perform better than the best single classifier in an ensemble classifier. This result motivates future work to add features to the dataset that may improve the performance of these models.

Therefore, machine learning models can be combined with other phishing detection techniques, such as list-based sample techniques, to improve performance. Besides, we will explore to propose and develop a new mechanism to extract new feature from the website to keep up with new techniques in phishing attacks.

IX. DATA AND CODE

To facilitate reproducibility of the research in this paper, all codes and data are shared at the GitHub repository :

X. ACKNOWLEDGMENT

This research is supported by Smart Supported by Land Co. Special thanks to Abed Farvardin for providing resources to this project and to Saeed Shahrivari for giving us the unique opportunity to run this great phishing detection project. This project also helped me a lot in my research and I learned so many new things. I'm really grateful to them.

REFERENCES

[1] FBI, “Ic3 annual report released.”
[2] APWG, “Phishing activity trends report.”
[3] V.B. et al, “study on phishing attacks,” International Journal of Computer Applications,2018.
[4] I.-F. Lam, W.-C. Xiao, S.-C. Wang, and K.-T. Chen, “Counteracting phishing page polymorphism: An image layout analysis approach,” in International Conference on Information Security and Assurance, pp. 270–279, Springer, 2009.
[5] W. Jing, “Covert redirect vulnerability,” 2017. Journal of Information Security and applications, vol. 22, pp. 113–122, 2015.
[6] P.Kumaraguru, J. Cranshaw, A. Acquisti, L. Cranor, J. Hong, M. A. Blair, and T.Pham, “School of phish: a real-world evaluation of antiphishing training,” in Proceedings of the 5th Symposium on Usable Privacy and Security, pp. 1–12, 2009.
[7] R. C. Dodge Jr, C. Carver, and A. J. Ferguson, “Phishing for user security awareness,” computers & security, vol. 26, no. 1, pp. 73–80, 2007.
[8] R. Dhamija, J. D. Tygar, and M. Hearst, “Why phishing works,” in Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 581–590, 2006.
[9] C. Ludl, S. McAllister, E. Kirda, and C. Kruegel, “On the effectiveness of techniques to detect phishing sites,” in International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 20–39, Springer, 2007.
[10] A. P. Rosiello, E. Kirda, F. Ferrandi, et al., “A layout-similarity-based approach for detecting phishing pages,” in 2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007, pp. 454–463, IEEE, 2007.
[11] S. Afroz and R. Greenstadt, “Phishzoo: Detecting phishing websites by looking at them,” in 2011 IEEE fifth international conference on semantic computing, pp. 368–375, IEEE, 2011.
[12] K.-T. Chen, J.-Y.Chen, C.-R. Huang, and C.-S. Chen, “Fighting phishing with discriminative keypoint features,” IEEE Internet Computing, vol. 13, no. 3, pp. 56–63, 2009.
[13] A. K. Jain and B. B. Gupta, “Phishing detection: Analysis of visual similarity based approaches,” Security and Communication Networks, vol. 2017, 2017.
[14] R. S. Rao and S. T. Ali, “A computer vision technique to detect phishing attacks,” in 2015 Fifth International Conference on Communication Systems and Network Technologies, pp. 596–601, IEEE, 2015.
[15] B. B. Gupta, N. A. Arachchilage, and K. E. Psannis, “Defending against phishing attacks: taxonomy of methods, current issues and future directions,” Telecommunication Systems, vol. 67, no. 2, pp. 247–267, 2018
[16] A. Karatzoglou, D. Meyer, and K. Hornik, “Support vector machines in r,” Journal of statistical software, vol. 15, no. 9, pp. 1–28, 2006.
[17] L. Breiman, “Random forests,” Machine learning, vol. 45, no. 1, pp. 5– 32, 2001.

[18] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class adaboost," Statistics and its Interface, vol. 2, no. 3, pp. 349–360, 2009.
[19] J. H. Friedman, "Stochastic gradient boosting," Computational statistics & data analysis, vol. 38, no. 4, pp. 367–378, 2002.
[20] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp. 785–794, 2016.
[21] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016.
[22] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Phishing websites features," School of Computing and Engineering, University of Huddersfield, 2015.