IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

"INTELLIGENT ASSOCIATION CLASSIFICATION TECHNIQUE AND COMPARATIVE ANALYSIS OF FEATURE-BASED MACHINE LEARNING" APPROACH FOR PHISHING WEBSITE DETECTION

[1]Ms. Minakshi Ramesh Marbade, [2]Dr. Rais Abdul Hamid Khan
[1]M.Tech. Data Science, [2]Associate Professor
Department of Computer Science & Engineering
G.H. Raisoni University, Amravati, Maharashtra, India

Abstract: Doing research in the field of cybersecurity makes significant year-on-year progress, phishing is one of the most commonly used DDoS attacks. Continuously increasing users and advances in web development, a very huge amount of the business world now solely depends on the internet. The number of cyberattacks and threats has also increased massively imposing monetary loss and concealment, Identity theft, self-assurance in online banking, and electronic acquisition, Phishing is one of those web attacks in which an attacker attempts to fraudulently acquire sensitive and personalized information from a victim by impersonating a sanctioned and trustworthy establishment. Again, phishing attack seems to be a problem for a while.
In this thesis, discover an efficacious and flexible malicious URL detection system with a rich set of features reflecting the different characteristics of phishing webpages and their hosting platforms, including features that are hard to forge by a troublesome. Using the Random Forests algorithm, our system enjoys the benefit of both high detection power and low error rates. based on carnal knowledge this is the first study to conduct such large-scale website/URLs computing, and classification experiments taking advantage of distributed points for feature collection. Experiment results exemplify that our system can be utilized for the automatic construction of blacklists by a blacklist provider.

Index Terms **- Phishing, Machine Learning, Classification, Approach, Cybercrime.**

I. INTRODUCTION

Nowadays, everyone uses the internet. With the ricochet of the Internet, a lot of sectors are using the Internet. A lot of traditional tasks and technologies as vast amounts of users or people have to access them. Now we have a lot of tasks that we can do online like shopping, banking, digital marketing, and so on. And some tasks have been almost entirely replaced by the Internet.

The Internet is evolving and growing. But, with the transition towards a world where many things are now being done online, many security concerns have grown exponentially too. Many users are still not aware of a lot of security problems and methods to deal with those issues. One of the important rising security concerns is phishing attacks and this research is dedicated to tackling the problem of phishing attacks.

Phishing is a social engineering attack where an assailant sends a counterfeit message that aims to make the most of the weakness found in system processes as caused by system users. e.g., Even though a system can be technically secure, unaware end users may leak their passwords. The assailant might ask users to update their passwords via a given Hypertext Transfer Protocol (HTTP) link, which would hindmost covenant the overall security of the system. such as technological threats and susceptibility e.g., Domain Name System (DNS) can be used by attackers to construct far more instigate socially-engineered messages. such as the use of consistent, but deride domain names can be far more alluring rather than using perspicuous domain names. Successively, phishing attacks are a scaly problem, and effective extenuation would require addressing both technical and human arguments [1].

Phishing uses social engineering tasks and techniques to trick users, such as creating fake and fraudulent websites that mimic the existing legitimate websites. During a classic phishing attack, a phisher sends a link confined in a message format to the victim. The link that leads the user to the cloned malicious webpage that looks similar to the original webpage does not intend to steal the user's sensitive information. Such a kind of website phishing attack has proven to cause a lot of financial loss to various organizations. This phishing attack can be obstructed by the abrogation of such harmful websites with the aid of "Phishing Detection" tools. Machine learning is the most powerful technique and tool which can make the detection of phishing websites a lot simpler. A machine learning-based tool will easily separate Phishing and Non-Phishing websites with the help of algorithms [11].

Phishing is the most prevalent cybercrime today. Phishing attacks can occur in a variety of domains, including online payment systems, webmail, finance, file hosting, cloud storage, and so on. Phishing attacks have been more common in the webmail and online payment sectors than in other industries. Phishing can be done in distinct ways like email phishing scams and spear-phishing, in which a user should be aware of the consequences and should not put their complete trust in a common security appliance.

Machine learning is the most efficient technique to detect phishing as it removes the drawbacks of various existing approaches.

1.1 The popularity of phishing attacks is as old as the Internet itself. The data available today show that as the Internet emerges, phishers also evolve their tactics and construct more elaborate attacks. This is backed by the following statistics, which capture the trends in phishing from the perspective of two renowned organizations: The APWG1 and Google.

1.2 The APWG Anti-Phishing Working Group (APWG) is an international coalition of more than 2200 members, which unifies the global response to cybercrime across industrial, governmental, law enforcement, and non-governmental organization region. Alongside other activities, their research efforts produce quarterly Phishing Activity Trend Reports, which analyze phishing attacks reported by the members of the APWG [2].

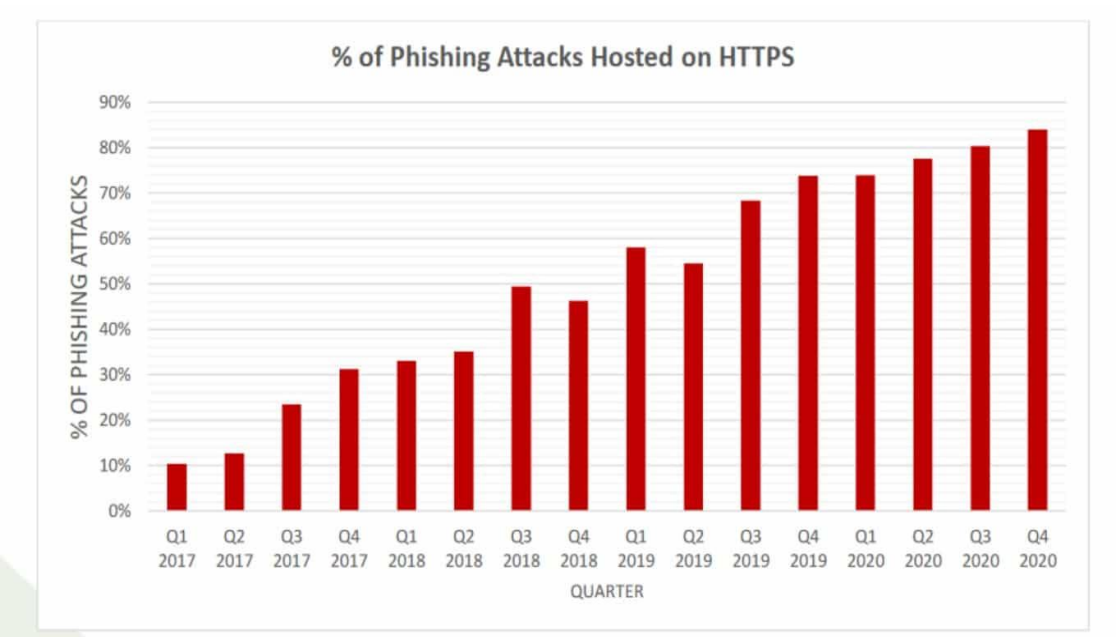

Figure 1: - Shows yearly figures of unique phishing websites detected by the APWG.[2]

II. DATASET

URLs from the benign website were collected from the Kaggle website [16], the total counting is 35300 which is the source dataset that is available on UNB. CA website, second will have valid CSV for website detection and this is an open-source file which is available on Phi's tank [17]. The services are provided into a set of phishing URLs in multiple formats like CSV, JSON, etc. So, we are using a CSV file for updating, we can download Phishing website details from phishtank.com then third we have a legitimate CSV file which extracted the feature of 5000 URLs which are randomly selected from 35300 datasets.

Then we have Phishing website and non-Phishing website data afterwards we can combine both data which is nearly 1000K so our model can run on that basis and we decide which URL is Phishing or which is non-Phishing. So, we have to combine the data in proportion like balance data. we have prepared 5000K Phishing URL data and 5000K Non-Phishing URL data So, in total we are using a 1000K URL dataset for model buildings.

III. OBJECTIVE

- The objective, which is the most indispensable fact in the proposed project, is to verify the legitimacy of the website by assimilating blacklisted URLs. To caution the user on a blacklisted website through a pop-up while they are upset to approach the URL and a platform for an individual to check and validate the integrity of any URL they want to access.
- The Objective is to apply Machine Learning techniques in the proposed approach to analyse the real-time URLs and produce effective results.
- The Objective is to implement the concept of the Blacklist approach, which is a familiar Machine Learning technique that can handle huge amounts of data.
- The objective is to develop a feature-based Approach to detect malicious URLs and alert users.

IV. METHODOLOGY

A phishing detection scheme or module detects a phishing attack by momentous that the domain is more similar to a known phishing domain or has been sizable as a phishing domain, and the or by detecting apprehensive network property and activity in that domain [3]. These modules can be implemented in a web browser on the client-side or the server-side through specific software. It can be more efficacious than training schemes or prevention schemes.

It doesn't rely on a gigantic amount of Internet users to educate themselves or go through increased training. Also, the extant authentication schemes can be used along with phishing detection schemes providing a much better user experience than complex user authentication. A phishing website disinterred by the phishing disclosure scheme can be blocked or the information superhighway user can be cautioned of the potential harm and that the host on the other side may not be legitimate Now, to make these solutions effective, the accuracy of the system is very imperative [3].

On one hand, if the detection system marks a phishing website as legitimate, the users would be left exposed despite the system being there in place to prevent users. Additionally, if the detection system marks a legitimate website as a phishing website, the author and the owner of the web page could suffer drastic damage potentially to their business.

4.1 Search Engine Based

This technique promotes features from web pages such as text, images, or URLs as a search string and determines the applause of the website. These are taken upon the inspection outcome of popular web search engines such as Google, Bing, and Yahoo. Most of the time, justifiable websites get back a substantial number of results and are ranked first, considering that phishing websites get no results and/or are not ranked at all [4][5]. The author used images, and Optical Scanning to excerpt text from those images, and then, uses Google Rank Algorithm to differentiate between legitimate and phishing websites.

4.2 Phishing Blacklist and Whitelist Based

In this mechanism, the unrecognized website is sluggish to a list of phishing and non-phishing websites (whitelisted). The phish tank fruitage and lend such a list of phishing websites onward with some details through the collective effort of a large commonwealth. These series are available to the developers out there who may wish to use the data in their applications or web pages for Anti-phishing purposes. Consequently, Alexa has a list of those whitelisted websites. The websites which are neither on those lists should be categorized as incredulous. The method is used quite often as it is computationally skillful. Widget like Google Safe Browsing employs a blacklist phishing revelation approach and IE Plug employs the whitelist method [12].

In general, the whole process of data science, or in this case, a machine learning experiment follows an all-purpose template, which can be outlined in the following six steps:

1] Defining a question/problem at hand.
2] Collection of extra raw data.
3] Pre-processing of the collected data.
4] Fundamental analysis.
5] Feature engineering.
6] Model training

1] Defining a question/problem at hand -

Start with what you admit -

When group members walk through the door at the starting phase of the meeting, what do they think about the position? There is a diversity of different ways to garner this information. In progression, people can be asked to write down what they know about the problem definition. The promotor can lead a bus session to try to bring out the greatest number of ideas. Recognize that a good facilitator will draw out everyone's assessment not only those of the more choral participants.

Decide what information is Lacking -

Information and Knowledge is the key to valuable decision-making. If we are swordplay children's hunger, do we know which children are hungry? When are they hungry - all the time, when the money has run out? If that's the case, our problem statement might be, "Children in our nation are oftentimes hungry at the end of the time because their parents' Paychex is used up too early."

2] Collection of extra raw data -

Bellows are the top six Data assortment mechanism

a. Interviews
b. Questionnaires and surveys
c. Observations
d. Records
e. Focus groups
f. Oral histories

3] Pre-processing of the collected data -

The goal of the pre-processing is to obtain a high-quality representation of the collected data. Depending on the attribute of the data/experiment, a variety of techniques are utilized to pre-process a dataset. Some datasets require data cleaning, dealing with missing values, feature selection, creating a train/test split, resampling, and many others.

This experiment has an advantage in that the data collection process results in genuinely raw data, which means we can be in charge of most of the steps. The first step after combining the datasets collected from multiple sources was to parse all of the URLs into isolated components to simplify manipulation with the dataset.

4] Fundamental analysis -

During the fundamental analysis phase, we attempt to identify significant features of our dataset and discover useful patterns. These can aid in generating ideas worthy of further investigation or, on the contrary, help discard wrong assumptions. Although descriptive statistics can also be used during fundamental analysis, during this experiment, a focus was put mainly on data visualization.

5] Feature engineering -

This category will describe the model used for extricating and engineering features that were used for this experiment.

URL occupying features [9][10] -

- Lengths of URL, subdomain, SLD, netblock, path.
- Digit count in URL, subdomain, SLS, netblock, and path.
- Domain count and path depth.

Natural language processing features -

The inclination for using this appearance is that phishing URLs tend to include specific vocabulary for some components of the URL (for example, words such as "boot," "camouflaged" or "account" tend to accomplish in the subdomain component).

Page based features -

The ground assumption is that phishing websites are usually hosted on non-reputable domains, meaning that page-based features should reflect the website's reliability. To extract these features, an open-source page ranking service Open PageRank (OPR) 5, was used. The OPR is a costless-to-use alternative to a more popular Alexa 6. It provides access to ratings and rankings of billions of websites through its REST API.

Domain squatting features -

Since phishers often try to imitate the names of famous brands and conservatory, we should try to quantify how suspicious is the URL from the perspective of domain/typosquatting. For this task, we extracted the top 500 highest ranked websites, according to OPR. After that, each subdomain and path segment were matched against the top 500 list. The highest achieved similarities for the path, subdomain, and SLD components were recorded as features. The similarity metric used for the string matching was normalized Lowenstein distance (also known as normalized edit distance).

6] Model training -

The machine learning models were trained using an open-source python machine learning library sci-kit- learn. During the hyperparameter control phase, 10-fold cross-validation was used for all of the algorithms. Depending on the nature of the hyperparameter space and time-complexity of each algorithm, either exhaustive grid search or randomized search was utilized.

- Decision Tree: -

The efficient time complexity of decision trees is usually linear. However, the domains for individual parameters can be theoretically infinite [13]. This means some heuristics were put in place to constrain the hyperparameter options to a smaller subspace. A factual study of decision tree hyperparameter tuning suggests the following for the hyperparameters of bother:
Split criterion: A function to measure the quality. The study suggests that for the vast majority of problems.

The usage of Gini scum versus entropy yields almost the same results. Therefore, Gini was used since it is faster to compute and is the default choice for the CART algorithm. The minimum number of samples required to split an internal node: empirical analysis shows that the optimal value usually falls in the range of around 2 to 40.

The minimum number of dossiers required at a leaf node accurate analysis shows that the optimal value is usually in the range of around 1 to 20. Based on these findings, an incidental search with 50 iterations was armed. The rest of the dimensions were fixed according to the default sci-kit-learn implementation.

The search united with cross-validation yielded the following best parameters:
The minimum sample leaf requires = 20
The Minimum sample splitting = 18

- Random Forest: -

The time complexity of random forests worsens with the growing number of trees generated in the forest, meaning an exhaustive grid search would be much more computationally expensive than with a decision tree, a random search was utilized as an alternative. The choice of model subspace for the construction of entity trees was based on the one used for the decision tree.

In addition, a random search also Based on the scrutiny of hyperparameter management strategies for random forests, is usually the best blueprint to use as many estimators as possible. However, the analysis suggests that the ensemble of the first 100 trees provides the most convergence [14]. This experiment settled with 200 estimators to achieve a judicious trade-off between performance and complexity.

The random search ran for 25 iterations and yielded a model with the following parameters:
The minimum sample leaf requires = 1
The Minimum sample splitting = 8
N estimator = 200

- K-nearest Neighbors: -

KNN is a very straightforward algorithm with equitable time complexity, even for medium-sized datasets like the one used for this experiment. The tuning phase subsists on a grid search for the Neighbor's parameter. This parameter represents the k in KNN, and the tested values ranged from 1 to 40 with steps equal to 1. The weight norm controls how the votes for predictions are cast: "uniform" considers all votes in the neighborhood of the sample as an adjunct while "distance" gives more weight to the vote of a closer Neighbor, for this experiment "distance" was worn. Finally, the classic Euclidean distance was set as a span metric. The results of the grid search showed that the optimal choice was k = 24.

- Support Vector Machine: -

Sickest-learn offers an optimized implementation of linear SVM, to take advantage of this feature, two separate random searches were conducted. The first is for the linear kernel only, and the second is for polynomial and radial basis operation kernels [15]. For the first search, values of polarization parameter C were drawn from a uniformly distributed continuous random variable in the range 10-3 to 5. As for the random search for polynomial and RBF kernels, apart from the C parameter, the inspected parameters were also kernel coefficient - gamma, and for the polynomial kernel, the choices of polynomial degree were 2 and 3. After 25 iterations of random search on the linear kernel and 10 iterations on the other kernels, the linear kernel generates the best results with the criterion C = 3.302.

- XG-Boost: -

XG-Boost is an enhanced dispersed gradient boosting library designed to be highly efficient, flexible, and portable. XG-Boost has been implemented in python, R, Ruby, Java, Julia, and many more programming languages. It is created with performance and speed in mind. It implements some countenance in the original gradient boosting algorithm that makes it an especially powerful tool. And its success has been reflected in structured data and tabular data. It is the most appointed algorithm to predict tabular training data. XG-Boost is a refined and customized rendition of Gradient Boosting to provide better performance and speed. The most important factor behind the success of XG-Boost is extensible in all plots.

The XG- Boost runs more than ten times more swift than prominent solutions on a single machine and ranges to billions of examples in disbursing or memory-limited settings. The scalability of XG-Boost is due to a handful of important algorithmic escalations. These innovations encompass a novel tree learning algorithm for handling sparse data; a theoretically persuasive weighted quantile sketch procedure enables handling instance weights in approximate tree learning. Parallel and distributed computing make learning faster which permits quicker model exploration.

More importantly, XG-Boost manipulates out-of-core computation and enables data scientists to process hundreds of millions of examples on a desktop. Finally, combining these techniques to make an end-to-end system that extends to even weightier data with the least amount of bundle resources is even more exciting [6].

❖ EVALUATION MATRIX

The data is labelled positive for phishing websites and negative for legitimate websites in our prediction. Based on this level, some of the terms used in the evaluation are given below:

By comparing the classification predictions with the actual categories of the emails, we can compute the numbers of true negatives (TN, correctly classified ham email) false negatives (FN, phishing email erroneously classified as ham), true positives (TP, correctly classified phishing email), and false positives (FP, ham email erroneously classified as phishing email). To evaluate the classifier performance, we compute the accuracy.

Accuracy: It is the rate at which the classifier correctly predicts the data. It can be calculated as:

Table 1: - Classification Matrix

	Classified of Phishing	Classified of Legitimate
Phishing	NP→P	NP→L
Legitimate	NL→P	NL→L

NP →Total number of Phishing websites
NL → Total number of legitimate websites

Performance evaluation parameters are as bellows:

- True Positive (TP): - Number of correct classifications of Phishing websites
  TP= (NP→P)/NP [7]
- True Negative (TN): - Number of correct classifications of ham websites
  TN= (NL→L)/NL [7]
- False Positive (TP): - Number of ham websites wrongly classified
  FP= (NL→P)/NL [7]
- False Negative (FN): - Number of wrong classifications of phishing websites
  FN= (NP→L)/NP [7]

Check the classification of webpages:

- Precision: The percentage of correct positive predictions is defined in precision.
  Precision=TP/ (TP+FP) [8]
- Recall: Percentage of positive prediction of positive labelled instances
  Recall=TP/ (TP+FN) [8]
- Accuracy: The percentage of correct prediction is defined as inaccuracy.
  Accuracy= (TP+TN)/(TP+TN+FP+FN) [8]
- F-Score: It measures a weighted average of true positive rate/recall and precision
  F=2*(precision*recall/precision+ recall) [8]

Table 2: - ML Model Accuracy

ML Model	Train Accuracy	Test Accuracy
Decision Tree	0.814	0.808
Random Forest	0.819	0.816
XG Boost	0.866	0.871
SVM	0.801	0.803

V. CONCLUSION

In this paper, we applied various machine learning algorithms Decision Tree, Random Forest, XG Boost, and SVM This paper aims to enhance detection methods to detect phishing websites using machine learning technology for the phishing detection sites have obtained results the results of this experiment show that we can detect phishing URLs with a precision of 96% (for the positive class) and classify URLs as benign or phishing with an accuracy of 95% using the XG boost algorithm.

The primary output of this thesis is an in-depth description of the possible approach to tackling the phishing detection problem without any web scraping. The secondary output is a standalone machine learning project that allows for future replications or modifications of the performed experiment.

REFERENCES

[1] Phishing Detection https://en.wikipedia.org/wiki/Phishing

[2] APWG report http://docs.apwg.org/reports/apwg_trends_report_q2_2018.pdf

[3] Goodman, Joshua T., et al. "Phishing detection, prevention, and notification." U.S. Patent No. 7,634,810.

[4] Huh, Jun Ho, and Hyoungshick Kim. "Phishing detection with popular search engines: Simple and effective." International Symposium on Foundations and Practice of Security. Springer, Berlin, Heidelberg, 2011.

[5] Dunlop, Matthew, Stephen Groat, and David Shelly. "Gold phish: Using images for content-based phishing analysis." 2010 Fifth international conference on internet monitoring and protection. IEEE, 2010.

[6] T. Chen and C. Guestrin, "XG-boost: A scalable tree boosting system," in Proceedings of the 22nd ACM signed International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.

[7] Almomani, A., Gupta, B. B., Atawneh, S., Meulenberg, A., & Almomani, E. (2013).” A Survey of Phishing Email Filtering Techniques”. IEEE Communications Surveys & Tutorials, 15(4), 2070–2090.

[8] Asha S Manek, D K Shamini, Veena H Bhat, P Deepa Shenoy, M. Chandra Mohan, K R Venugopal, L M Patnaik. "ReP-ETD: A Repetitive Pre-processing technique for Embedded Text Detection from images in spam emails", 2014 IEEE International Advance Computing Conference (IACC)

[9] Arun Kulkarni, Leonard L. "Phishing Websites Detection using Machine Learning", International Journal of Advanced Computer Science and applications, 2019

[10] Aron Balm, Brad Wardman, Thamar Solorio and Gary Warner (2010),” Lexical feature-based phishing URL detection using online learning”, 3rd ACM Workshop on Security and Artificial Intelligence.

[11] Purvi Pujari, M. Chaudhari (2018) “Phishing Website Detection using Machine Learning: A Review”

[12] Jain, A.K.; Gupta, B. Comparative analysis of feature-based machine learning approaches for phishing detection. In Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India,16–18 March 2016; pp. 2125–2130.

[13] http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/

[14] http://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/

[15] https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html

[16] https://www.Kaggle.com

[17] www.phishtank.com