Phishing Detection from URLs Using Ensemble Techniques

Vijay Venkatesh M
Department of Computer Science
University of Madras
Guindy, Chennai - 600025, India
vijayvenkatesh2212@gmail.com

Pradeep N
Department of Computer Science
University of Madras
Guindy, Chennai - 600025, India
pradeep1098729@gmail.com

Abstract* - *The growing number of phishing attacks is one of the top concerns of security researchers today. The traditional approach which is used to identify phishing websites is inconsistent and not reliable. URL phishing attacks cannot be prevented via cryptographic methods, because the attackers can use various means to trick a user into clicking on the malicious link. This results in compromising sensitive information such as passwords, smart card pins, etc. Recent research progress has shown how machine learning methods are capable of detecting malicious websites. In this paper, we will look at the comparison between traditional machine learning models and ensemble techniques. The experiment is conducted on a dataset with 30 features containing 11056 phishing and benign website links in total. This experiment demonstrates that the ensemble algorithms perform better compared to other traditional machine learning methods. Out of which light gradient boosting algorithm obtains the highest accuracy compared to other machine learning models.

Index Terms* – *Machine learning, Ensemble learning, Phishing, Supervised learning.

I. INTRODUCTION

The growth of the internet is tremendous in recent times and it's the world's largest growing dynamic network. There are over 5 billion internet users each day. Internet is ubiquitous today, we use web services for a range of activities such as knowledge sharing, social communication, and conducting various financial activities which include buying, selling, and money transfers as well as unwanted content such as phishing, drive-by download, host drive-by exploits, and spam [20].

Considering the at-most activity of users on the internet, Phishing is one of the most common types of social engineering attacks that target the weakness created by the system user[2]. It creates an illusion of integrity in the URLs and proceeds to persuade the user enough to take action which compromises their sensitive information such as passwords, identity details, social security numbers, or any such valuable information [1]. According to IBM's 2022 Cost of Data Breach Report, Phishing becomes the costliest cause of compromising information. While compromised credentials continued to be the leading cause of security breaches (19%), phishing was the second largest (16%) and most expensive cause, resulting in an average cost of $4.91 million for responding organizations [6]. CISCO states that in 2021 the probabilistic percentage of at least one individual clicking a phishing link from an organization is about 86% [7]. In Q2 2022, APWG observed a total of 1,097,811 phishing attacks, a new record and the worst quarter for phishing APWG has ever observed. The average requested amount for BEC attacks on remittances in Q2 2022 was $109,467, up from $91,436 in Q1 2022. The healthcare and transportation industries suffered from an increase in ransomware attacks in Q2. Mobile-based fraud has increased with smishing and vishing increasing in Q2 [8]. Figure 1 illustrates that in the 1st quarter of 2022, Statista Research Department found that the most frequent targets were finance and SaaS sites.

Numerous approaches have been employed to filter out phishing websites. Such as server-side filters, authentication, protection, and user education [5]. Although there are some distinctive characteristics in these attacks, which leads the researchers to deploy machine learning techniques. Since ML is powerful when it comes to quantizing a problem to find patterns and relations in them. In this paper, we will be looking at the pre-processing methods of URLs. Followed by the comparison of traditional machine learning models and ensemble methods in Phishing detection classification. Which is accompanied by K-Fold cross validation of accuracy.

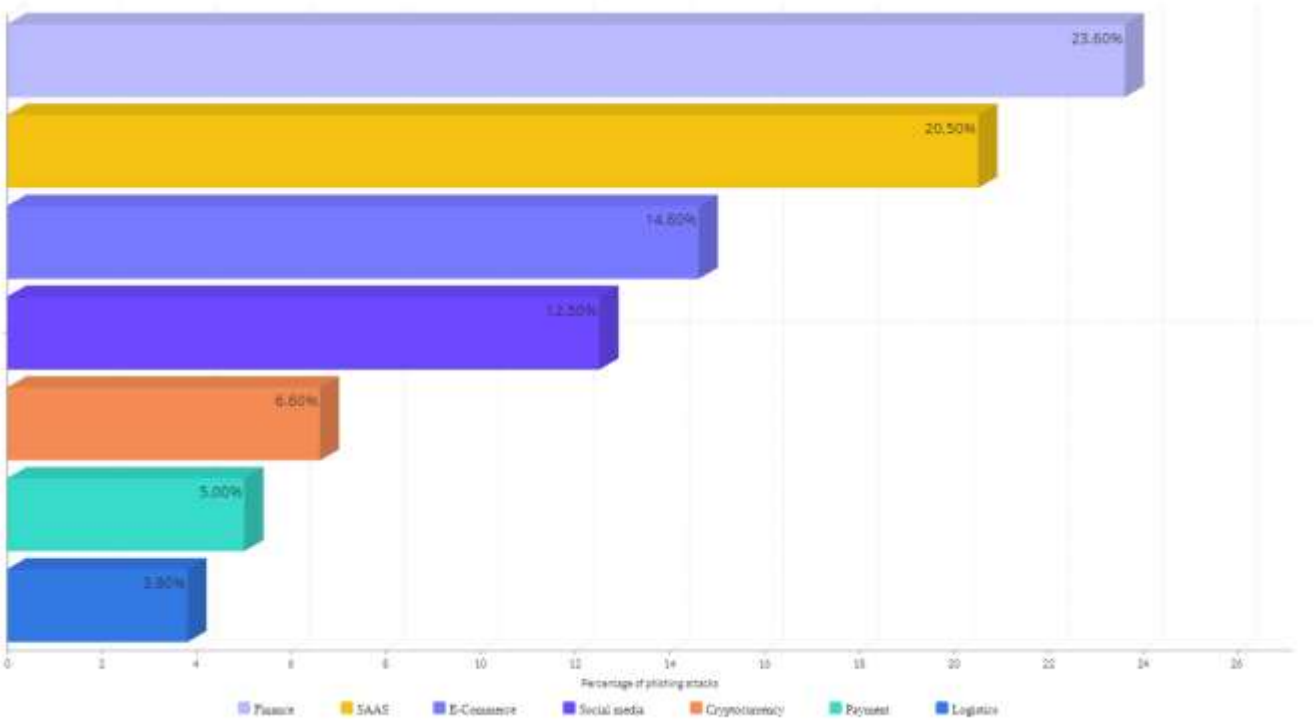

Figure 1. Most Targeted Industries, 1Q 2022

II. Dataset Summary

A. Data Collection

The process of collecting malicious and benign URLs with well-labeled formats is indeed challenging. Fortunately, PhishTank an open-source website provides live API for malicious URLs [21]. Benign websites are collected from the University of New Brunswick [12].

B. Feature Engineering

The most challenging part of phishing detection is pre-processing the raw URL into subsequent independent features to feed into the model. Though some of the preprocessed datasets are readily available in certain public repositories to train the model. In order to evaluate a newly suspicious URL, one must structure it into the required format.

Fortunately, the pre-processing technique which is employed in the paper is well-researched by the respective authors [11]. The URLs are labeled as phishing by the following criteria.

1. **IP Address:** Presence of IP Address in the domain.
2. **Long URL:** Suspiciously long URLs to hide doubtful part.
3. **Tiny URL:** Using URL shortening services to make the URL looks legitimate.
4. **"@" Symbol:** URL's consisting of the symbol "@".
5. **Redirection by "//":** Existence of "//" within the URL path, especially after 7 indices (the default for "HTTPS" is 6).
6. **Adding Prefix or Suffix:** The symbol dash (-) is rarely used in legitimate URLs.
7. **Multi Sub Domain:** Which are indicated by the domain part containing more than one dots.
8. **SSL State:** If the SSL state is not available.
9. **Domain Registration Life-Period:** Domain registration period with less than a year (short life-time period).
10. **Favicon:** If the graphic image loaded from an external source domain.
11. **Using Non-Standard Port:** If the port is of a preferred status.
12. **HTTPS:** Having https token in URL from a suspicious issuer.
13. **Request URL:** If the percentage of Request URL is greater than 61%.

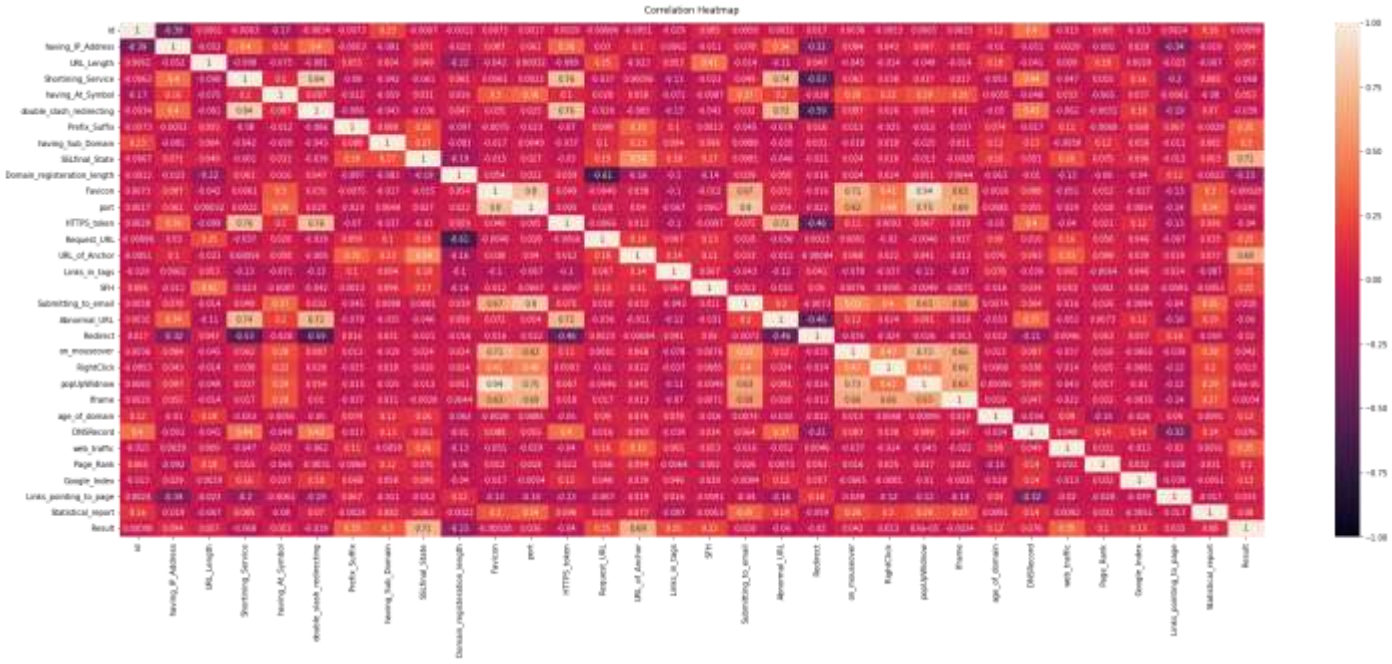

Figure 2. Correlation Heatmap

14. **URL of Anchor:** If the percentage of Anchor tags <a> of URL is greater than 67%.
15. **Links In Tags:** If the percentage of Links in < Meta > "," < Script > " and " < Link>" is greater than 81%.
16. **Server Form Handler:** If the SFH different from the domain name of the webpage.
17. **Submitting Information to E-mail:** Using "mail()" or "mailto:" function to submit user information.
18. **Abnormal URL:** host name is not being included in URL.
19. **Website Forwarding:** If the redirect pages is more than 4.
20. **Status Bar Customization:** If the "onMouseOver" state changes the status bar.
21. **Disabling Right Click:** If the right click is being disabled.
22. **Using Pop-up Window:** If the new Popup windows contains text fields.
23. **IFrame Redirection:** URL containing iframe tag.
24. **Domain Age:** Having a domain of age less than a month.
25. **DNS Record:** If the website has no DNS record.
26. **Website Traffic:** Rank of the website is less than 100,000.
27. **Page Rank:** If the PageRank is below 0.2, The scale of page rank is from 0 – 1.
28. **Google Index:** If the website is not indexed by google.
29. **Number of Links Pointing to Page:** If there's no link pointing to the webpage (98% of phishing dataset items have no links pointing to them).
30. **Statistical-Reports Based Feature:** If the host belongs to top phishing Ips / Domains.

The above mentioned 30 features are used to train/test the models. Figure 2 Shows the dependency/relations of all the features in the dataset.

III. Experimental Results

A. Traditional ML Models

Logistic Regression: Logistic regression is a method of modeling the likelihood of a discrete outcome given an input variable. The most popular logistic regression models a binary outcome, something that can take two values such as 0 or 1. This is possible because of the sigmoid activation function.

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (1).$$

Even if the input value is very high or very low, the sigmoid activation function will contract the values of the range 0 to 1. Logistic regression is an advantageous analysis method for classification problems, where you are trying to determine if a recent sample fits best into a category [10].

KNN Classifier: K – Nearest Neighbor algorithm works on the basic principle of measuring the distance between two points in

space and finding relations with respect to the closeness of points. The standard distance metric used in KNN is Euclidean Distance. The number 'K' of neighbors is chosen initially. Then, by taking the K Nearest Neighbor of unknown data point w.r.t distance, we find the category where most of the neighbors lie in. Then the unknown data point is assigned to that category.

SVM Classifier: Support Vector Machine (SVM) is widely preferred in supervised classification problems because of its effectiveness in mapping higher dimensional spaces [16]. The data points are plotted in n-dimensional space (n is the number of features). The classification criteria are to find the optimal split of the classes using a hyperplane. Two vectors are constructed using PAC – theory and a non-zero Lagrange multiplier (to determine the threshold of vectors).

Then the splitting hyperplane is introduced such that it is equidistant from the two supporting vectors. The SVM algorithm also has a sneaky trick in classifying complex data, which is called the kernel trick. The kernel SVM is a function that takes low-dimensional input and transforms it into higher-dimensional space [14]. Thus, it is easier to classify more complex data.

Naive Bayes Classifier: It is a classification model which is used to discriminate different objects based on certain features. It works on the principle of Bayes theorem.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2).$$

A – hypothesis
B – evidence

By using Bayes theorem, the probability of A happening can be found given that B has already occurred. With a 'naïve' assumption of conditional independence between every pair of features given the value of the class variable. Hence the name naïve bayes. In spite of their oversimplified assumption, they have worked really well in some cases. One of the greatest advantages of this method is that they are extremely fast.

Decision Tree Classifier: It is a rule-based approach algorithm, which is used to classify numerical and categorical variables. It forms a tree-like structure, where every node has a condition for splitting into subsequent nodes. The split condition is heavily dependent on the purity of distribution in classes. The impurity of the split is calculated using the Gini index or entropy methods [15]. Each node in the tree is constructed w.r.t result from the Gini index. The tree grows until it meets the base case, which might be based on the impurity ratio or the number of nodes.

Table 1. Performance Measure of Traditional ML Models

Model Name	Accuracy	5 – Fold Cross Validation	Standard deviation
Logistic Regression	92.28%	93.11%	0.56
KNN Classifier	93.6%	93.77 %	0.31
SVM	92.19%	92.93 %	0.66
Kernel SVM	94.57%	95.01 %	0.38
Naive Bayes	60.53%	59.56 %	0.61
Decision Tree	96.29%	95.88 %	0.62

B. Ensemble Methods

In general, ensemble techniques seek better outcomes by combining the results of multiple models. The two main types of ensemble methods are as follows.

- Bagging: It is a process of running diverse groups of models parallelly by varying the training data, then taking the optimal solution out of it.
- Boosting: It is a sequential process, where the output of a previous model is fed as input to the next model. So, the prediction from the prior model is corrected/improved by the later one.

Random Forest: It is a bagging ensemble approach, where many decision tree classifiers are fed with various sub-sample of the dataset. The predictive measure is by averaging all individual decision trees and taking the majority vote. The main advantage of this method is to control overfitting and improve accuracy.

XGBoost Classifier: Extreme Gradient Boosting is a scalable, distributed gradient-boosted decision tree (GBDT). It provides a parallel tree boosting and works very well with GPUs. Some of the unique features of XGBoost are regularization, Handling sparse data, Weighted quantile sketch and Cache awareness.

Extra Tree Classifier: It is almost similar to Random Forest, but the node splits are not based on the Gini index or Entropy measure. The splits are purely random [19]. The randomness doesn't come from bootstrapping of data, but rather comes from the random splits of all observations.

Ada Boost Classifier: It is an adaptive boosting technique. It is based on decision trees but with only one level, these decision trees with a single split are called Decision Stumps [17]. The algorithm gives equal weights to all the data points. Then it assigns higher weights to points that are wrongly classified. Now all the points which have higher weights are given more

importance in the next model. It will keep training the models until and unless a low error is received.

Gradient Boosting: In gradient boosting, each predictor corrects its predecessor's error. In contrast to Ada-boost, the weights of the training instances are not tweaked, instead, each predictor is trained using the residual errors of predecessor as labels [18].

Light Gradient Boosting: LGB is an extension of the gradient boosting algorithm by adding a type of automatic feature selection as well as focusing on boosting examples with larger gradients. This can result in a dramatic speedup of training and improved predictive performance. There are two important key ideas in this algorithm's working principle.

- **GOSS:** Gradient-based One-Side Sampling, is a modification to the gradient boosting method that focuses attention on those training examples that result in a larger gradient, in turn speeding up learning and reducing the computational complexity of the method.
- **EFB:** Exclusive Feature Bundling, or EFB for short, is an approach for bundling sparse mutually exclusive features, such as categorical variable inputs that have been one-hot encoded. As such, it is a type of automatic feature selection.

Together, these two changes can accelerate the training time of the algorithm by up to 20 times [13].

Table 2. Performance Measure of Ensemble Models.

Model Name	Accuracy	5 – Fold Cross Validation	Standard deviation
Random Forest	95.86%	95.74 %	0.53
XG Boost	94.39%	95.01 %	0.48
Extra Tree	95.99%	96.11 %	0.39
Ada Boost	93.42%	93.81 %	0.76
Gradient Boost	94.57%	95.05 %	0.46
Light Gradient Boosting	96.26%	96.39 %	0.27

IV. Conclusion

In this paper, we have reproduced the feature engineering techniques and observed the outcomes. The dataset consists of 6157 benign and 4898 malicious URLs. We have implemented six traditional ML models along with six ensemble techniques. In accordance with Table 1 and Table 2. It is indeed clear that the ensemble models are in general great in performance measures. We have also verified the accuracy with 5 – Fold cross-validation for certainty. The performance of Extra Tree and Light Gradient Boosting is really good compared to other ensemble models. But Extra Tree Classifier does not perform bootstrap aggregation like in the random forest. Thus, the nodes split are random splits and not optimal splits. It mainly focuses on accuracy, thus giving low variance. Most of the ensemble learners have high computational costs. But LGB works faster by reducing the computational complexity through GOSS. It also works better over a period of time, because it uses EFB for automated feature selection. Thus, the LGB model seems the best fit for URL Phishing Detection.

As for future work, we will conduct extensive experiments by using different sets of parameters to obtain the highest possible detection accuracy. In addition, we also plan to propose a different approach to feature engineering to extract important variables from the raw URL.

X. Data And Code

To facilitate reproducibility of the research in this paper, all codes and datasets are shared at this GitHub repository: https://github.com/vijaysr4/ANVAYA-paper-presentation.

References

[1] David Lacey, Paul Salmon, Patrick Glancy, "Taking the Bait: A Systems Analysis of Phishing Attacks", Procedia Manufacturing, vol 3, pp 1109-1116, 2015.

[2] M. Khonji, Y. Iraqi and A. Jones, "Phishing Detection: A Literature Survey", in IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091-2121, Fourth Quarter 2013.

[3] A. A. Zuraiq and M. Alkasassbeh, "Review: Phishing Detection Approaches," 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS), 2019, pp. 1-6.

[4] Sadiq, A., Anwar, M., Butt, R. A., Masud, F., Shahzad, M. K., Naseem, S., & Younas, M. "A review of phishing attacks and countermeasures for internet of things-based smart business applications in industry 4.0", Human Behavior and Emerging Technologies, vol 3, pp 854– 864, 2021.

[5] Shahrivari, V, Darabi, M. M, & Izadi M, "Phishing Detection Using Machine Learning Techniques", arXiv, 2020.

[6] "IBM data breach report", https://www.ibm.com/reports/data-breach, 2022.

[7] "CISCO's 2021 report", https://umbrella.cisco.com/info/2021-cyber-security-threat-trends-phishing-crypto-top-the-list, 2021.

[8] APWG, "Phishing activity trends report", https://apwg.org/trendsreports/, 2022.

[9] Mohammad, Rami, Fadi Thabtah, and T. L.McCluskey, "Phishing websites dataset", 2015.

[10] Thomas W. Edgar, David O. Manz, “Research Methods for Cyber Security”, Chapter 4 – Exploratory Study, Editor(s): Thomas W. Edgar, David O. Manz, pp 95-130, 2017.

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

[12] “University of New Brunswick - Datatsets”, https://www.unb.ca/cic/datasets/url-2016.html, 2016.

[13] Ke, Guolin and Meng, Qi and Finley, Thomas and Wang, Taifeng and Chen, Wei and Ma, Weidong and Ye, Qiwei and Liu, Tie-Yan, “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”, in “Advances in Neural Information Processing Systems”, vol 30, 2017.

[14] Jain, A.K., Gupta, B.B. “PHISH-SAFE: URL Features-Based Phishing Detection System Using Machine Learning”. In: Bokhari, M., Agrawal, N., Saini, D. (eds) Cyber Security. Advances in Intelligent Systems and Computing, vol 729. Springer, Singapore, 2018.

[15] S. Sivagama Sundhari, "A knowledge discovery using decision tree by Gini coefficient," in International Conference on Business, Engineering and Industrial Applications, 2011, pp. 232-235, 2011.

[16] Zhang, Y. Support Vector Machine Classification Algorithm and Its Application. In: Liu, C., Wang, L., Yang, A. (eds) Information Computing and Applications. ICICA 2012. Communications in Computer and Information Science, vol 308. Springer, Berlin, Heidelberg, 2012.

[17] Chengsheng, Tu & Huacheng, Liu & Bing, Xu, “AdaBoost typical Algorithm and its application research”.,MATEC Web of Conferences. 139, 2017.

[18] Natekin, Alexey & Knoll, Alois, “Gradient Boosting Machines, A Tutorial”, Frontiers in neurorobotics, 2013.

[19] Geurts, P., Ernst, D. & Wehenkel, L. “Extremely randomized trees”, Mach Learn 63, pp 3–42, 2006.

[20] A. Alswailem, B. Alabdullah, N. Alrumayh and A. Alsedrani, "Detecting Phishing Websites Using Machine Learning," in 2nd International Conference on Computer Applications & Information Security (ICCAIS), 2019, pp. 1-6, 2019.

[21] PhishTank, Operated by Cisco Talos Intelligence Group, https://phishtank.org/index.php.