Study and Analysis of Machine Learning Models for detection of Phishing URLs

Mr.Shreyas Desai
Computer Science and Technology
Department of Technology
Kolhapur, Maharashtra, India
shreyasdesai3013@gmail.com

Mr.Sahil Salunkhe
Computer Science and Technology
Department of Technology
Kolhapur, Maharashtra, India
sahilsalunkhe29@gmail.com

Dr. Mrs. Rashmi Deshmukh
Computer Science and Technology
Asst.Prof.Department of Technology
Kolhapur, Maharashtra, India
rvm_tech@unishivaji.ac.in

***Abstract*— The increase in use of the internet has led to the increase in usage of many e-commerce, social media and a lot of other websites where it is required by the user to give his/her sensitive credentials while using. Stealing user's sensitive information without consent by deceiving him about the genuineness of the website is called phishing. Most times the users aren't aware that they are visiting a website which is trying to steal the information and unknowingly they give away a lot of personal information. This leads to internet fraud and a lot of cybercrimes. To avoid this, we have proposed a phishing detection method with minimal number of features and higher accuracy as possible. We have used a number of machine learning models and compared their performance based on some performance metrics. We have taken phishing URLs from the PhishTank dataset and legitimate URLs from Alexa Dataset.**

Keywords— phishing, URL, feature extraction

I. INTRODUCTION

After the digitalization has begun almost all the services are at our doorstep most of them being Mailboxes, Ecommerce, Banking, social media, Food Services etc. Majority of people started using these services at an increasing pace. These services ask you to authorize yourself in order to use them, this involves sharing your personal information to them. As for the legitimate services this is not needed but for the forged ones this may even lead to a critical situation. This is when the phishing begins, it's been around for more than 30 years, and a large number of users are deceived every year.

Phishing is a type of social engineering attack which is frequently used to steal customer's credentials. Fraudsters trick customers into typing credentials, credit card or maybe bank account numbers on those fraudulent phishing sites which they host on some hosting services providers. As at least one user gets trick and click on such web sites the DNS server also registers such fraudulent websites on their database. As it comes beneath the social engineering act, it varies from user to user on how they get lured in. The fundamental vector a fraudster can attempt is to first speak about the similar interest the user has followed by a phishing activity like mail or website. The phishing attacks are so easy to drag out relying upon the user and have one of the worst results feasible. One may become completely bankrupt. Not every time it is about stealing credentials some may even ask to click on a malicious link which can implant a backdoor in a company's system which till detected allows the fraudster to get the whole insider, the possibilities are endless and needs to be stopped.

Phishing can be defined as luring the users into giving out their personal information to the forged services acting like legitimate ones. It is considered as a cybercrime and there are various types of phishing attacks out there like Vishing, Whaling, Spear Phishing, and Email Phishing. This paper specifically focuses on website phishing and implements the idea of phishing URL detection using ML.

Phishing came from the word "phish" which in simple terms is known as 'the fraud' and is the serious issue of data security. There are multiple technologies developed over the period of time to stop phishing attacks like the blacklist method which maintains such a database of phishing URLs and blacklists them on the DNS. The Heuristic Based Method uses feature extraction to detect the phishing Web sites but it's also the easiest one to bypass once an attacker knows features used. Another being the content based anti-phishing which uses visual similarities to identify the contents of phishing websites from legitimate websites by analysing the similarity of the contents.

As of late, there have been a few examinations attempting to take care of phishing issues. In this paper we have implemented different machine learning models for detection of phishing URLs based on the features of the URL such as length, paths in URL, subdomains present, etc. We have used a custom dataset which is created using URLs from the PhishTank dataset for phishing URLs and Alexa dataset for legitimate URLs. At the end we have compared the performance of different models on the dataset based on performance metrics like accuracy, precision etc.

II. LITERATURE REVIEW

Abdelhakim Hannousse , Salima Yahiouche [1] proposed a novel combination of features that are extracted solely from the URL. They evaluate their performance by using a dataset which consists of active phishing attacks and compare it with Google Safe Browsing (GSB). We can also see four models (SVM, RF, LR, KNN) approach which uses only nine

features based on the lexical properties of URLs and produces an accuracy of 99.57%[3]. Similarly, Md. Faisal Khan and B. L. Rana [5] who anticipated a 10 features approach and the models being used with accuracy LSTM 98.67%, DNN 96.33%, CNN 97.23%. After analysing few more papers we found out that even DT is best suited to find out such URLs as we can see in "Phishing URL Detection Using Machine Learning'' [9] where they compared four models LR, DT, RF, SVM and by comparing four models on the basis of six factors (absolute error, root mean squared error, sensitivity, training, time, cohen_kappa_score, roc_auc_score) among 16 factors.

Some of the papers even implemented a hybrid solution of multiple approaches (heuristic features, visual features and various approaches feeding these distinct features to machines) [7]. Seok-Jun Bu and Sung-Bae Cho [10] used a hybrid of neural networks and logic programming which has the best prediction level even for 0-day phishing by using CNN-LSTM based triplet network.

The KNN too seems to be effective against such kinds of URLs as Tsehay Admassu Assegie[11] proposed KNN model with accuracy of 85.06% using 106 observations for testing the model on phishing detection. Using accuracy metrics for experimental results shows that the model is effective against phishing attacks. There are even seven different algorithms (DT, Adaboost, K-star, kNN (n = 3), RF, SMO and Naive Bayes) approaches used to increase the accuracy of the detection system [8].

III. Methodology

A great deal of the works that we have reviewed as of recently uses URL based features along with the HTML content inside the website to correctly distinguish whether a website is phishing or not. The proposed work has been depicted in the figure below.

A. Proposed Work

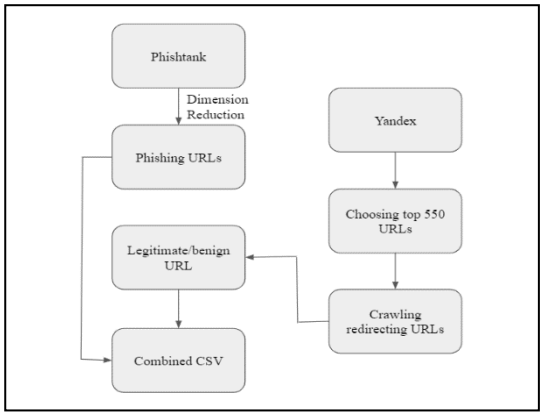

Fig. 1 Proposed System Architecture

As shown in Fig.1 we have started with the phishing URL dataset from the PhishTank.org website. After the dataset is downloaded, we have checked the shape of the dataset (features present and number of rows). We have reduced the number of elements in the feature vector from the already given dataset to the ones which we require. Dataset of top 1 million websites is collected from the Kaggle data servers. With this, domain names of the top 1million websites are obtained.

After this a python script is executed to crawl the top 550 website domains present in the list to find redirecting URLs present on each of the domains. A list is obtained after crawling through all these website domains containing all the redirect URLs which can then be treated as a set of benign/legitimate URLs. Lastly, after obtaining two different CSV files, one containing phishing URLs and another containing Legitimate URLs, both the CSVs are combined and shuffled. The model training part is done on this "combined.csv" file.

B. Traditional Methods

Regardless of whether the site is phishing or legitimate is decided on URL features of the website. Different features from the URL are extracted based on its structure. Various features can be used for detecting phishing of URLs. The traditional methods for phishing detection were Blacklist Method, Heuristic based model and visual similarity-based methods. These techniques are as follows-

Blacklist Method

Blacklist method is a traditional method which uses a pre-maintained database consisting of phishing URLs which are identified and updated regularly. In this method the browser simply checks if the URL is already present in the database or not. The drawback of this method is that it cannot keep up with the continuous creation of the new phishing URLs as keeping the database updated every second is not possible. Also, it cannot automatically identify if the URL is phishing or not.

Heuristic Based Model

This is an updated version of the blacklist method based on the pre-fed algorithm where the system is able to automatically identify if a URL is phishing or not. But this system is easy to bypass once the attacker knows the algorithm which is used to detect phishing URL.

Visual Similarity

Commonly, a phisher attempts to trick a user by using visual similarities from legitimate websites and causing the user to accept that he is visiting a genuine site. In the visual similarity method, the algorithm attempts to analyse the visuals from the phishing websites with that of legitimate websites. Downside of this framework is that it takes more time to compare images and regardless of whether there is a little change in the pictures, the algorithm may fail.

Machine Learning Based Approach

Aside from the conventional methodology stated in the above methods we can use machine learning approaches to deal

with identifying a phishing URL. The different features we can use while detecting using machine learning are HTML content-based features, NLP based detection and URL based detection. In our project we have attempted to minimise the overload on the system by minimising the features used for faster training and trying to achieve as much high accuracy as possible.

IV. Results and Experimentation

The implementation of the project is done with a system having Processor - i5 9th generation, with 8GB RAM and Secondary Memory - 1TB HDD. Operating system is Windows 11 and experimentation is carried out with IDE - Jupyter Notebooks

A. Dataset Creation

For the implementation; we have created a custom dataset. This dataset is created with the help of PhishTank and Alexa. PhishTank is used for the collection of phishing URLs whereas Alexa is used for the collection of legitimate URLs. As mentioned in the proposed work, for the creation of the dataset we have downloaded the phishing URL dataset from the PhishTank (~4200 URLs). Pre-processing is carried out by eliminating unwanted columns from the dataset.

Alexa dataset contains 1 million URLs. As Alexa dataset records the top sites all over the world, most of the websites incorporate names like “googl.com”, “amazon.com”, “facebook.com”, etc. These website URLs are not useful for training our data because we can’t extract any useful features from the previously mentioned websites. To avoid this, we have executed a python script which visits the first 550 websites and crawls their webpages to discover the redirecting URLs present on the webpage. With this we have got a large set of URLs (~17700) with multiple features and which can be used to train our model.

B. Feature Extraction

We have used URL based features for training our model so that the overhead on the model is reduced. There are no. of features which can be used based on URL from which it can be deduced whether it is phishing or not. From the features available, we have used the below given features.

Table 1: Features Table

No.	Feature	Values in Dataset
1	Length of URL	Integer
2	Whether IP address present or not	0 and 1
3	Whether any shortening service is used or not	0 and 1
4	Special character counts in URL	Integer
5	Presence of ‘ // ’ in URL other than “[http/https:]//” in URL	0 and 1
6	Phish Hints in the URL	Integer
7	Common Term count in the URL	Integer
8	Digit Ratio	Float
9	URL depth	Integer
10	Abnormal Subdomain	0 and 1

C. Training data and Comparison

For the implementation we have used six different machine learning models. Comparison of the learning models is based on the accuracy score, precision, recall and F-1 score on the training as well as testing data.

XGB (eXtreme Gradient Boosting)

XGBoost is an algorithm which has emerged recently, dominating the applied machine learning for structural or tabular data. XGBoost is an extended version of gradient boosting decision trees which improves speed and performance.

Logistic Regression

Logistic Regression is a supervised machine learning algorithm which is used to predict categorical dependent variables based on independent variables. Logistic regression works best when there are 2 discrete values i.e. True or False, 0 or 1, Yes or No, etc. It is used for classification problems.

RFC (Random Forest Classifier)

Roughly explaining, **Random Forest Classifier** is a collection of number of decision Trees. These decision trees work on various subsets of the dataset and the average of the trees is taken to improve the predictive accuracy. This model gives more accurate predictions as compared to a single decision tree. The higher the number of trees in RFC, the higher the accuracy and lesser overfitting.

Decision Tree

Decision Tree Classifier is a supervised learning algorithm which is best suited for classification problems. In Decision trees, there are leaf nodes and decision nodes. Decision nodes are nodes which make decisions and branches into leaf nodes. Leaf Nodes do not have branches and they represent the results of the tree.

SVM (Support Vector Machines)

SVM is a supervised learning algorithm which can be used for regression as well as classification problems. In SVM, the main objective is to find a hyperplane in an N-dimensional

space that distinctly classifies the data points. The dimensions of the hyperplane depend upon the number of input features.

KNN (K-Nearest Neighbours)

KNN is a widely used regression and classification supervised learning algorithm. The working of KNN is simpler to understand. In KNN we simply pick a data point and put it in the graph with a feature matrix as the axes. We then simply set the no. of K (no. of neighbours) around the data point. What KNN does is, it classifies the data point based on the similarities of the data points which are already present. Then it checks the count of the data points in each category and assigns the new data point to the category with the maximum number of neighbours.

D. Model-Metric Analysis

While doing an analysis of different machine learning models we take into consideration various factors such as its accuracy, time taken for the model to execute, how much hyper parameter tuning can be done to make the model work at its best, etc. We have taken some of these factors to check the performance analysis of the models. We will be doing the analysis of the Models based on the confusion matrix for different data parts in the dataset.

Table 2 shows a comparison of different models based on the above-mentioned performance metrics on both, training as well as testing data.

Table 2: Comparison of Models based on different performance metrics

	Accuracy		**Precision**		**Recall**		**F1 Score**	
Model	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
XGBooster	94.98	88.02	92.15	83.13	95.35	86.58	93.73	84.82
Logistic Regression	76.06	75.94	67.08	66.32	72.14	71.15	65.92	68.92
Random Forest Classifier	94.81	88.32	91.88	83.13	95.20	87.26	93.51	85.14
Decision Tree	95.00	86.91	91.17	80.99	96.36	85.69	93.69	83.27
Support Vector Machine	91.10	85.53	86.70	79.72	91.01	83.58	88.80	81.60
K Nearest Neighbors	93.28	83.74	92.28	77.75	91.30	81.80	91.79	79.38

From Table 2, we can say that except the logistic regression model, most of the models work efficiently on the training data. But while using a machine learning model in a real time scenario, we have to take into consideration how well the model can perform on unseen data. Hence from the above table, we will focus more on the columns having metric analysis for test data. From Fig.2, we clearly see that the XGBooster model and the Random Forest Classifier have the best test data accuracy among the given machine learning models. The precision and recall of these models are also noteworthy as based on the precision and recall, we come to know how much of the predicted values are false negative or false positive, as these values have a greater effect on how well our model works.

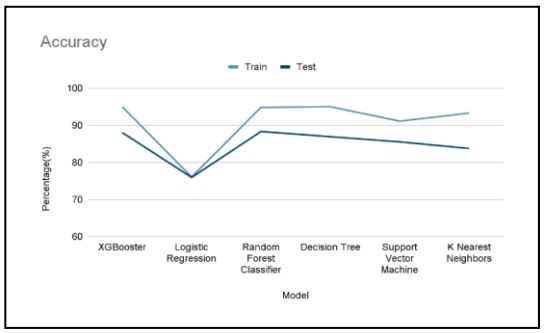

Fig. 2 Train and Test data Accuracy for various MLs

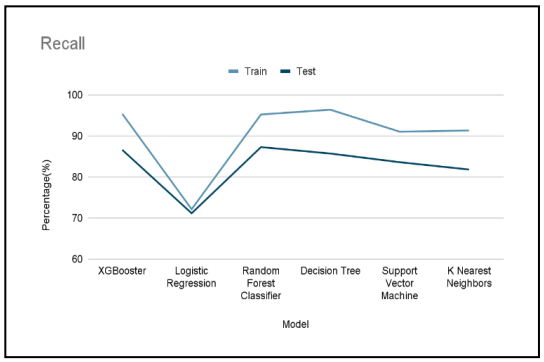

Fig. 3 Train and Test data Recall for various MLs

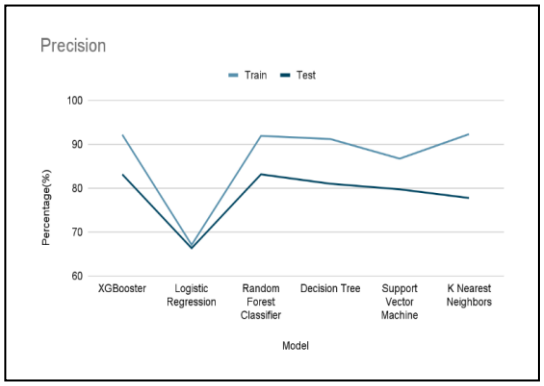

Fig. 4 Train and Test data Precision for various MLs

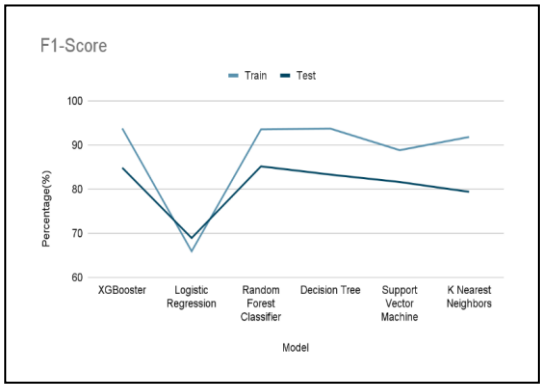

Fig. 5 Train and Test data F1 score for various MLs

V. Conclusion

In this paper, we began with an outline about phishing attacks present in real-time scenarios and traditional methods present to detect and invalidate them. We then studied different approaches used for detection of the phishing URLs. We have tried to overcome some of the drawbacks of the existing system. For the implementation we have created a custom dataset, and then some features are extracted from it. Then different Machine Learning models are applied to the dataset. Finally, we have compared these models based on different metrics namely Accuracy, Precision, Recall, F1-Score etc. Experimentation shows that the Random Forest Classifier or the XGBoost model can be used to identify a phishing URL using a minimum number of features.

References

[1] Abdelhakim Hannousse , Salima Yahiouche. "Towards benchmark datasets for machine learning based website phishing detection: An experimental study." *Engineering Applications of Artificial Intelligence*, vol. 104, 2021. 0952-1976.

[2] Andrei Butnaru, Alexios Mylonas, Nikolaos Pitropakis. "Towards Lightweight URL-Based Phishing Detection." *Multidisciplinary Digital Publishing Institute*, 2021.

[3] Brij B. Gupta, Krishna Yadav, Imran Razzak, Konstantinos Psannis, Arcangelo Castiglione, Xiaojun Chang. "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment." *Computer Communications*, vol. 175, 2021, pp. 47-57. 0140-3664.

[4] C Vineeth Krishna. "Identification of Phishing Urls Using Machine Learning." *Journal of Physics: Conference Series*, 2021. 1770-012009.

[5] Md. Faisal Khan and B. L. Rana. "Detection of Phishing Websites Using Deep Learning Techniques." *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 10, 2021. 3880-3892.

[6] Mehmet Korkmaz, Ozgur Koray Sahingoz, Banu Diri. "Detection of Phishing Websites by Using Machine Learning-Based URL Analysis." *IEEE Xplore*, 2020.

[7] Nikhil K, Dr. Rajesh D S, Dhanush Raghavan. "Phishing Website Detection Using ML." *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 7, no. 4, 2021, pp. 194-198. 2456-3307.

[8] Ozgur Koray Sahingoz, Ebubekir Buber, Onder Demir, Banu Diri. "Machine learning based phishing detection from URLs." *Expert Systems With Applications*, vol. 117, 2019, pp. 345-357. 0957-4174.

[9] Preeti, Rainu Nandal, and Kamaldeep Joshi. "Phishing URL Detection Using Machine Learning." *Springer*, 2021.

[10] Seok-Jun Bu and Sung-Bae Cho. "Integrating deep learning with first-order logic programmed constraints for zero-day phishing attack detection." *IEEE Xplore*, 2021.

[11] Tsehay Admassu Assegie. "K-Nearest Neighbor Based URL Identification Model for Phishing Attack Detection." *Indian Journal of Artificial Intelligence and Neural Networking (IJAINN)*, vol. 1, no. 2, 2021. 2582-7626.