Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7750424

DETECTING PHISHING WEBSITES USING RANDOM FOREST ALGORITHM

Dr.V.Rameshbabu[1], S. Mohan[1], Dr.D.Usha[1]
Neha Kumari[2], Pushpamitra.P[2],Raafia Fathima.Y[2]
[1]Professor, Department of CSE, Dr.MGR Educational and Research Institute,Chennai
[2]Final Year B.Tech-CSE, Dr.MGR Educational and Research Institute,Chennai
[1]drvramesh25@gmail.com, mohan.cse@drmgrdu.ac.in, usha.cse@drmgrdu.ac.in
[2]nehakumari217315@gmail.com,
pushpaprabhu44@gmail.com,raafiafathima1112@gmail.com

Abstract: Phishing is a technique used by hackers or attackers to trick users into inserting their sensitive credentials such as usernames, passwords, and credit card details into an unauthenticated entity, such as such as website. In this type of attack, unauthenticated entities disguise themselves as authentic and trusted entities. Sources are fooled by the look and feel of the fake website which closely resembles the legitimate website. Phishing is a type of cybercriminal that tries to get important or confidential information from users, this is usually achieved by creating a fake website that mimics a legitimate website. Phishing attacks use a variety of techniques such as link manipulation, filter avoidance, site spoofing, hidden redirects, and social engineering. Scammers use social engineering to trick victims or set up fake websites to steal personal and organizational information such as account IDs, usernames, and passwords.

Keywords: Phishing, Personal information, Machine Learning, Malicious links, Phishing domain characteristics.

I. INTRODUCTION

Detecting any phishing site is indeed a complex and dynamic matter involving many factors and criteria. Phishing sites are fake websites created by malicious actors to look like a real website. Phishing is sending emails to users posing as legitimate businesses with the intent of defrauding or tricking users into revealing personal information that will be used for identity theft. The impact is a breach of information security through a breach of confidential data and the victim may suffer monetary or other losses. The latest reports have shown that most of the phishing attacks are online scams targeting the financial, business and payment industries. Phishing sites are a very complex issue to understand and analyse as they combine technical and social issues, with the most common method being to create a fake website that mimics a legitimate website.

The internet has become an integral part of our lives, but it also creates opportunities for malicious activities like anonymous phishing. Scammers try to trick their victims through social engineering or create fake websites to steal information like account IDs, usernames, passwords from individuals and organizations. There is no known magic formula for a complete solution.

In this article, we compared the results of several machine learning methods to predict phishing websites.

II. LITRATURE SURVEY

(Dutta AK *et al*., 2021) has discovered phishing websites using machine learning technique. In this study, the author proposed a Uniform Resources Locators (URL) detection technique based on machine learning identify malicious and legitimate websites. It has achieved better accuracy and F1—score with limited amount of time. The experiments' outcome shows that the proposed method's performance is better than the recent approaches in malicious URL detection.

(Selvakumari M *et al.,* 2021) found this technique has been effective, the attacker can still be able to rat run the blacklist system by modifying the required characters in the URL in order to make the system believe that it is legitimate, and it makes the system not to identify it as a phishing website. Each algorithm will give its evaluated accuracy after all the algorithms return its result. Each algorithm is compared with other algorithms to see which provides the high accuracy percentage. To identify the websites which are fraud, this paper will discuss the machine learning and deep learning algorithms and apply all these algorithms on the dataset and the best algorithm having the best precision and accuracy is selected for the phishing website detection. The methodology they discovered is a powerful technique to detect the phished websites and can provide more effective defences for phishing attacks of the future.

(Ahmed *et al.,* 2016) has proposed "Real time detection of phishing websites." Their main objective of this attack is to steal the sensitive information from the users. The attacker creates a `shadow' website that looks similar to the legitimate website. This fraudulent act allows the attacker to observe and modify any information from the user. This paper proposes a detection technique of phishing websites based on checking URLs of web pages. The proposed solution is able to distinguish between the legitimate web page and fake web page by checking the URLs of suspected web pages. URLs are inspected based on particular characteristics to check the phishing web pages. The detected attacks are reported for prevention.

(Jain *et al.,* 2016) has proposed "Comparative analysis of features based machine learning approaches for phishing detection. This paper presents a comprehensive analysis of Phishing attacks, their exploitation, some of the recent machine learning based approaches for phishing detection and their comparative study. It provides a better understanding of the phishing problem, current solution space in machine learning domain, and scope of future research to deal with Phishing attacks efficiently using machine learning based approaches.

(Fu, A.Y *et al.,* 2006) Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Movers Distance (EMD) IEEE transactions on dependable and secure computing. An effective approach to phishing Web page detection is proposed, which uses EMD to measure Web page visual similarity. They first convert the involved Web pages into low resolution images and then use color and coordinate features to represent the image signatures. They use

EMD to calculate the signature distances of the images of the Web pages. They trained an EMD threshold vector for classifying a Web page as a phishing or a normal one. Large-scale experiments with 10,281 suspected Web pages are carried out to show high classification precision, phishing recall, and applicable time performance for online enterprise solution. They also compare our method with two others to manifest its advantage and also built up a real system which is already used online and it has caught many real phishing cases.

III. PROPOSED METHOD

Detecting phishing sites, the use of machine learning is a suggested mechanism intended to provide undue security. The research version of the device is built using know-how strategies of precise records such as variable identity, it is based on unbiased variables. The log viewer is then run to preview the recordings. The version is purely based on the previous data set, where the set of rule study records and separately trained algorithms are used for higher comparisons. Overall performance metrics are calculated and compare. This chapter introduces various UML schemas for detecting phishing websites and describes their various functions.

System Modules

The system module consists of 4 steps. They:

• Dataset — A textual dataset containing information about phishing websites.

• Pre-processing- A pre-processing step to separate data and labels.

• Training- Here we train a machine learning classifier. A random forest classifier is used to classify the entered site as deceptive.

• Experiment- During the testing phase, the classifier predicts whether the new website URL is a phishing URL

IV. MODULE DESCRIPTION

- Data Pre-processing
- Data Analysis of Visualization
- Evaluating Algorithm
- Deployment Using Flask

4.1 Data Pre-processing

Step 1: Data set collection is the first step of data pre-processing in machine learning. Machine building and development learning model, the system must first obtain the relevant data set.

Step 2: Importing all important libraries is the second step of data pre-processing in machine learning. Built-in Python libraries can perform specific data pre-processing tasks.

Step 3: The system will import the data set. By importing dataset, python has created different modules which help us to import external data in different file formats into python program.

Step 4: In data pre-processing, it is very necessary to correctly identify and handle missing values, there are 2 ways to handle missing data:

- Delete a specific row
- Calculate the average

Step 5: Encrypt categorical data. Categorical data refers to information that has specific categories in the data set.

Step 6: Feature Scaling: Feature scaling marks the end of data pre-processing in Machine Learning. It is a method to normalize the independent variables of a data set in a particular range.

4.2 Evaluating Algorithms

It is important to compare the performance of multiple different machine learning algorithms consistently.

The four different algorithms which are compared:

4.2.1 Random Forest Algorithm

4.2.2 Decision Tree Classifier

4.2.3 Logistic Regression Algorithm

4.2.4 Naive Bayes Algorithm

The K-fold cross-validation process is used to evaluate each algorithm, it is important to be configured with the same random origin to ensure that the same splits of the training data are performed and each Algorithms are evaluated in the same way. Before this comparison algorithm, let's build a machine learning model using the Scikit-Learn implementation libraries.

Prediction result by accuracy:

Logistic regression algorithm also uses a linear equation with independent predictors to predict a value.

True Positive Rate (TPR) = TP / (TP + FN)
False Positive Rate (FPR) = FP / (FP + TN)

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non- defaulters.

Accuracy calculation:

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

Precision:

Precision = TP / (TP + FP)

Recall:

Recall = TP / (TP + FN)

General Formula:

F- Measure = 2TP / (2TP + FP + FN

F1-Score Formula:

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Random Forest Algorithm

Random Forest is a popular machine learning algorithm related to supervised learning techniques. It can be used for both classification and regression problems in ML. It is based on the concept of synchronous learning, which involves combining multiple classifiers to solve complex problems and improve model performance.

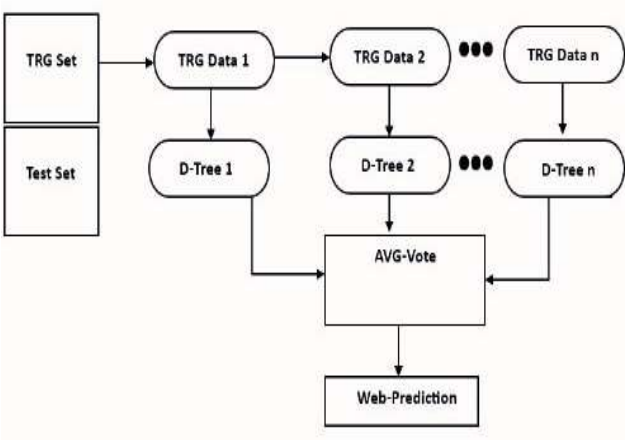

Figure 4.1 Workflow of Random Forest Algorithm

Random forest or random decision forest is a synchronous learning method for classification, regression and other tasks, which works by building an infinite number of decision trees at training time and generating class as the mode of classes (classification) or the predicted mean (regression) of individual trees. Random decision forests overcome the habit of overfitting decision trees into their training set. Random forest is a type of supervised machine learning algorithm based on synchronous learning. Synchronous learning is a type of learning where you join multiple types of algorithms or the same algorithm over and over again to train a more robust predictive model. The random forest algorithm combines several algorithms of the same type, i.e., several decision trees, resulting in a forest of trees, hence the name "random forest". RF algorithms can be used for both regression and classification problems.

Workflow of Random Forest Algorithm

Random Forest works in two phases, the first stage is to generate a random forest by combining N decision trees and the second stage is to make a prediction for each tree generated in the first stage.

The working process can be explained in the steps and diagrams below:

Step 1: Randomly select K data points from the training set.

Step 2: Build a decision tree associated with the selected data points (subset).

Step 3: Choose the number N for the decision trees that the system wants to create.

Step 4: Repeat steps 1 and 2.

Step 5: For the new data points, find the predictions from each decision tree and assign the new data points to the category that received the majority of votes.

Decision tree algorithm

Decision tree algorithms are a type of supervised learning algorithm. This works for continuous and categorical output variables. Decision trees build classification or regression models from tree structures. The decision tree is grown incrementally while dividing the data set into smaller and smaller subsets. A decision node has two or more branches and a leaf node represents a classification or a decision. The highest decision node in the tree corresponding to the best predictor is called the root node. Decision trees can handle both categorical and numerical data.

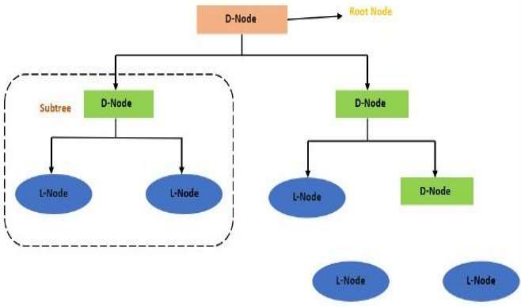

Figure 4.2 Workflow of Decision Tree Algorithm

When implementing decision trees, the main problem that arises is how to choose the best attribute for the root node and for the child nodes. So, to solve such problems, there is a technique called Attribute Selection Measurement or ASM. By this measure, we can easily choose the best attribute for the nodes of the tree. A tree has many real-life analogues and has been found to have influenced a wide field of machine learning, including classification and regression. In decision analysis, a decision tree can be used to visually and unambiguously represent decisions and the decision-making process. There are two common techniques for ASM, which are:

o Receive information
o Gini index

Decision tree algorithm workflow

Step 1: Start the tree with a root node, called S, containing all the data.
Step 2: Find the best attribute in the data set using the attribute selection index (ASM).

Step 3: Divide S into subsets containing the possible values of the best attributes.
Step 4: Create a decision tree node containing the best attribute.
Step 5: Recursively generate a new decision tree using subsets of the dataset created in step 3. Continue this process until you reach the stage where you can no longer rank the nodes and call the last node a leaf node.

4.2.3 Logistic regression algorithm

Logistic regression is one of the most popular machine learning algorithms and belongs to supervised learning techniques. Used to predict a categorical dependent variable given a set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore, the results should be categorical or discrete values. It can be yes or no, 0 or 1, true or false, etc. However, instead of giving exact values like 0 and 1, it provides probability values between 0 and 1.

Logistic Regression equations are given below:

The equation of the straight line can be written as:

y=b0+b1x1+b2x2+b3x3+…. +bnxn

In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y): y ; 0 $for\ y = 0$, $and\ infinity\ for\ y = 1$ $1 - y$

But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$\log\left[\frac{}{1-y}\right] = b0 + b1x1 + b2x2 + b3x3 + \cdots + bnxn$$ Equation 1

Workflow of Logistic Regr

Step-1: For I =1 to k

Step-2: For each training data instance di:

Step-3: Set the target value for the regression to

$$z_i \leftarrow \frac{y_j - P(1 \mid d_j)}{[P(1 \mid d_j) \cdot (1 - P(1 \mid d_j))]}$$ Equation 2

Step-4: Initialize the weight of instance dj to P(1|dj).(1-P).(1|dj)

Step-5: Finalize a f(j) to the data with class value (zj) & weights (wj)

4.2.4 Naïve bayes algorithm

Naïve Bayes algorithm is a supervised learning algorithm, based on Bayes theorem and used to solve classification problems. Naïve Bayes Classifier is one of the simplest and most efficient classification algorithms for building fast machine learning models that can make fast predictions. It is a probabilistic classifier, which means it makes predictions based on the probability of an object.

Bayes' Theorem:

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes' theorem is given as:

Naïve Bayes Classifier Algorithm

P(A|B) =P(B|A) P(A)/P(B) Equation 3
where,

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

Workflow of Naïve Bayes Algorithm

Step 1: Calculate the prior probability for given class labels.

Step 2: Find Likelihood probability with each attribute for each class.

Step 3: Put this value in Bayes Formula and calculate posterior probability.

Step 4: See which class has a higher probability, given the input belongs to the higher probability class.

Detection of URL

The implementation of the algorithms is done to generate whether the URL is phishing, legitimate or suspicious and the following figure 7.1 represents the webpage for detection.

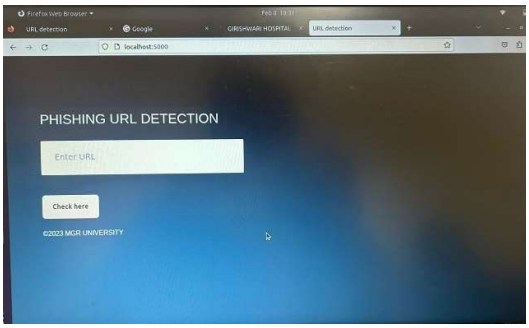

Figure 4.3 Detection of the URL

The given URL is phishing, legitimate or suspicious can be predicted by using the features of the website.

V. RESULT AND DISCUSSION

```
print("Accuracy result of Random Forest Classifier is:",accuracy.mean() * 100)
LR=accuracy.mean() * 100
```

```
    accuracy                           0.89       406
   macro avg       0.88      0.86      0.87       406
weighted avg       0.89      0.89      0.89       406


Confusion Matrix result of Random Forest Classifier is:
 [[192   2  17]
 [  2  25   4]
 [ 18   2 144]]

Sensitivity :  0.9896907216494846

Specificity :  0.9259259259259259

Cross validation test results of accuracy:
[0.86715867 0.90405904 0.88929889 0.90740741 0.9037037 ]

Accuracy result of Random Forest Classifier is: 89.43255432554325
```

Figure 5.1 Accuracy of Random Forest Algorithm

The final value can be calculated by taking the average of all values predicted by all trees in the forest, or in the case of classification problems, each tree in the forest predicts the category to which the new record belongs. The sensitivity and specificity of the random forest classifier were 0.98 and 0.92. The system did the cross-validation test results for accuracy and yielded 90% with the lowest false positive rate. Finally, a new record was awarded to the category that won the majority of votes. The system achieved detection accuracy of 89.43% using the random forest algorithm. Furthermore, the results show that the classifiers give better

performance when we use more data as training data.

```
Classification report of Decision Tree Classifier Results:

              precision    recall  f1-score   support

          -1       0.86      0.91      0.88       211
           0       0.77      0.77      0.77        31
           1       0.87      0.80      0.83       164

    accuracy                           0.85       406
   macro avg       0.83      0.83      0.83       406
weighted avg       0.86      0.85      0.85       406


Confusion Matrix result of Decision Tree Classifier is:
 [[192   2  17]
 [  4  24   3]
 [ 28   5 131]]

Sensitivity :  0.9896907216494846

Specificity :  0.8571428571428571

Cross validation test results of accuracy:
[0.85239852 0.87453875 0.88191882 0.88518519 0.88888889]

Accuracy result of Decision Tree Classifier is: 87.6586032526992
```

Figure 5.2 Accuracy of Decision Tree Algorithm

This process continues on the training set until the stopping condition is satisfied. It is built using the top-down recursive divide-and-conquer method. All attributes must be classified. Attributes at the top of the tree have more impact on classification and are determined using the concept of information acquisition. The sensitivity and specificity of the decision tree classifier were 0.98 and 0.85. The system performed a cross-validation test for accuracy and yielded 87% with the lowest false positive rate. The system achieves detection accuracy of 87.6% when using decision tree algorithm.

```
Classification report of Logistic Regression Results:

              precision    recall  f1-score   support

          -1       0.82      0.91      0.86       211
           0       0.56      0.16      0.25        31
           1       0.81      0.82      0.81       164

    accuracy                           0.81       406
   macro avg       0.73      0.63      0.64       406
weighted avg       0.80      0.81      0.80       406


Confusion Matrix result of Logistic Regression is:
 [[191   1  19]
 [ 14   5  12]
 [ 27   3 134]]

Sensitivity :  0.9947916666666666

Specificity :  0.2631578947368421

Cross validation test results of accuracy:
[0.79335793 0.82287823 0.8302583  0.84814815 0.84814815]

Accuracy result of Logistic Regression is: 82.85581522481891
```

Figure 5.3 Accuracy of Logistic Regression Algorithm

The result is measured with a binary variable. The goal of logistic regression is to find the best fit model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent variables (predict or explain).

The sensitivity and specificity of the logistic regression algorithm were 0.99 and 0.26. The system performed a cross-validation test for accuracy and yielded 82% with the lowest false positive rate. The system achieved detection accuracy of 82.8% using the algorithm.

```
Classification report of Naive Bayes Results:

              precision    recall  f1-score   support

          -1       0.83      0.88      0.85       211
           0       0.36      0.16      0.22        31
           1       0.79      0.80      0.80       164

    accuracy                           0.80       406
   macro avg       0.66      0.62      0.62       406
weighted avg       0.78      0.80      0.78       406


Confusion Matrix result of Naive Bayes is:
 [[186   3  22]
 [ 13   5  13]
 [ 26   6 132]]

Sensitivity :  0.9841269841269841

Specificity :  0.2777777777777778

Cross validation test results of accuracy:
[0.79335793 0.80811808 0.83763838 0.82592593 0.82592593]

Accuracy result of Naive Bayes is: 81.81932485991527
```

Figure 5.4 Accuracy of Naïve Bayes Algorithm

The goal is to try to rank the data by maximizing P(O|Ci)P(Ci) using the following Bayesian probability theorem (where O is the Object or set in the dataset and "i" " is the index of the class). Tests with high specificity (high true negative rate) are more useful when the result is positive. It is easy and fast to predict the class of the test dataset. The sensitivity and specificity of the naive Bayesian algorithm were 0.98 and 0.27. The system performed cross-validation for accuracy and yielded 81% with the lowest false positive rate. The system achieved a detection accuracy of 81.8% using the naive Bayesian algorithm.

VI. CONCLUSION

Therefore, detecting phishing websites by URL using a supervised machine learning algorithm reduces the chances

of being hacked. The analysis process begins with data cleaning and processing, missing values, exploratory analysis, and finally model construction and evaluation. This article presents a method of establishing rules for converting the input characteristics of a web page into an output that reveals the characteristics of a web page, whether it is deceptive or not. You can find better accuracy - higher accuracy, score on public test suite. The system performed a detailed documentation review of the detection of phishing sites. Thus, the system can say that

one machine learning tree classifier is better than another. This app can help you to know if phishing site is predictable or not. The system achieved detection accuracy of 89.43% using the random forest algorithm with the lowest false positive rate. The results also show that the classifier performs better when it uses more data as training data.

REFERENCES

[1] Dutta, A.K. (2021), "Detecting Phishing Websites using Machine Learning Technique" PloS one, Vol.16, No.10, p.e0258361.

[2] Selvakumari, M, Sowjanya, M, Das, S. and Padmavathi, S, (2021) "Phishing Website Detection using Machine Learning and Deep Learning Techniques. In *Journal of Physics: Conference Series.*, Vol. 1916, No. 1, p. 012169, IOP Publishing.

[3] Ahmed, A.A. and Abdullah, N.A, (2016) "Real Time Detection of Phishing Websites". In *2016 IEEE* 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vol. 22, No. 1, pp. 1-6 IEEE.

[4] Jain, A.K. and Gupta,(2016) B.B, "Comparative Analysis of Features Based Machine Learning Approaches for Phishing Detection. In 2016 3rd international conference on computing for sustainable global development (INDIACom), Vol. 32, No. 1, pp. 2125-2130.

[5] Fu, A.Y., Wenyin, L. and Deng, X, (2006) "Detecting Phishing Web Pages with Visual Similarity Assessment based on Earth Mover's Distance (EMD)", IEEE transactions on dependable and secure computing, Vol. 3, No. 4, pp.301-311.

[6] Dhamija, R. and Tygar, J.D, (2005) "The Battle against Phishing: Dynamic Security Skins". In Proceedings of the 2005 symposium on Usable privacy and security, Vol. 21, No. 6, pp. 77-88.

[7] Adida, B., Hohenberger, S. and Rivest, R.L.,(2005) "Lightweight Encryption for Email", Vol. 34, No. 5, pp. 5-9.

[8] Islam, M. and Chowdhury, N.K, (2016) "Phishing Websites Detection using Machine Learning Based Classification Techniques". In International Conference on Advanced Information and Communication Technology, Chittagong, Bangladesh, Vol. 10, No. 9, pp. 4393-4402.

[9] Tahir, M.A.U.H., Asghar, S., Zafar, A. and Gillani, S, (2016) "A Hybrid Model to Detect Phishing-sites using Supervised Learning Algorithms". In 2016 International Conference on Computational Science and Computational Intelligence (CSCI) Vol. 9, No. 1, pp. 1126-1133.

[10] Huang, H., Zhong, S. and Tan, J, (2009) "Browser-side Countermeasures for Deceptive Phishing Attack". In 2009 Fifth International Conference on Information Assurance and Security, Vol. 1, No. 12, pp. 352-355.