International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 6 June 2022, pp: 226-232 www.ijaem.net ISSN: 2395-5252

Phishing Website Detection

Dr.N. Jaya Lakshmi [1], Bh. Sai Surendra [2], B. Rupa [3], B.K.S. Chaitanya [4], K. Nayana Chowdary [5]

[1]*Associate Professor, Raghu Institute of Technology, Visakhapatnam, Andhra Pradesh*
[2, 3, 4] *Student, Raghu Institute of Technology, Visakhapatnam, Andhra Pradesh*

Submitted: 01-06-2022 Revised: 05-06-2022 Accepted: 08-06-2022

ABSTRACT: Phishing websites is a type of social engineering attack which is used widely by phishers and unauthorised people to manipulate or make the users believe that the website which they are using is a legitimate website. The ultimate goal is steal the data or personal information without the consent of the respective user. The phishers make the website look exactly same visually and make the phished website work mostly exact as the original website. So, to assist and aid the people in detecting the phished website we use machine learning here.

KEYWORDS: Phishing, Gradient Boosting, Machine learning, Feature Importance, Exploratory Data Analysis (EDA), Classification Report.

I. INTRODUCTION

Now-a-days, technology is growing at a very fast pace. From a 2-year old child to 90year old people, everyone irrespective of the age are using the technology and accessing, using the services of the internet through applications, websites. But many of the people who are using these are unaware of the cyber security practices and are an easy target to become victim of cyber attack.

Phishing became one of the dangerous cyber attacks as it may lead to great amount of losses. It is not a herculean task to create a fake website which looks mostly same as the legitimate website. Cyber security experts may identify the fake sites but not users can identify them and those people are more prone to be attacked. Top people and management such as CEO, CFO, CTO etc in the technological field are also easily manipulated as phishing attacks exploit the weaknesses in the users.

Phishing websites are made in such a way that cyber security experts also sometimes misjudge them as legitimate website as they can't process all the factors associated with it to determine if a website is legitimate or a phished website accurately. So, this is where machine learning comes into play as it can determine phished or not more accurately.

The main objective in doing this study is

1. To develop a machine learning model that classifies website into legitimate and phished in a most accurate way.
2. Aiding and alerting users by detecting phishing more accurately and efficiently.
3. To get the knowledge and hands-on experience in new technologies.

II. RELATED WORKS

According to several recent papers, phishing study in the context of cyber security and practices is not yet complete, and it remains an ongoing study topic, as evidenced by the following:

1. **Fadi Thabtah et al. Intelligent rule based phishing websites classification**:

Phishing is described as the art of echoing a website of a creditable firm intending to grab user's private information such as usernames, passwords and social security number. Phishing websites comprise a variety of cues within its content-parts as well as the browser-based security indicators provided along with the website. Several solutions have been proposed to tackle phishing. Nevertheless, there is no single magic bullet that can solve this threat radically. One of the promising techniques that can be employed in predicting phishing attacks is based on data mining, particularly the `induction of classification rules' since anti-phishing solutions aim to predict the website class accurately and that exactly matches the data mining classification technique goals. In this study, the authors shed light on the important features that distinguish phishing websites from legitimate ones and assess how good rule-based data mining classification techniques are in predicting phishing websites and which classification technique is proven to be more reliable.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 6 June 2022, pp: 226-232 www.ijaem.net ISSN: 2395-5252

2. **S. Marchal et al., (2017)** proposed this technique to differentiate Phishing website depends on the examination of authentic site server log knowledge. An application off-the-Hook application or identification of phishing website. Free, displays a couple of outstanding properties together with high preciseness, whole autonomy, and nice language-freedom, speed of selection, flexibility to dynamic phish and flexibility to advancement in phishing ways.

III. EXPERIMENTAL DATASET

The choice of dataset plays an important role in any machine learning project. Oftentimes, there exists a number of potentially suitable datasets available, each with their own advantages and disadvantages, and the decision to choose one dataset over another can significantly influence the course of the project. Cyber security experts have been trying for a long time to understand phishing and what differentiates between legitimate and phished website.

The data set is in text file and csv file which provides the following resources that is used as inputs for model building :

A collection of website URLs for 11054 websites. Each sample has 32 website parameters and a class label identifying as

- -1 is Phishing
- 0 is Suspicious
- 1 is Legitimate

1. The dataset for a ".txt" file is with no headers and has only the column values.
2. We can add the header manually if we are using '.txt' file. We used '.csv' file, the column names were added and given.

Using the IP Address
If the domain has an IP address → Phishing (-1)
Else → Legitimate (1)

Long URL
If URLlength<54 → Legitimate (1)
If URLlength≥54 and ≤75 → Suspicious (0)
Else → Phishing (-1)

Using URL shortening services
If used URL shortener → Phishing (-1)
Else → Legitimate (1)

URL has "@" symbol
URL had @ symbol → Phishing (-1)
Else → Legitimate (1)

Redirections
The index of the last occurrence of "//" >7 --> (-1)
Else → Legitimate(1)

Adding (-) to the domain
domain had (-) symbol → Phishing (-1)
Else → Legitimate (1)

Sub domain and Multi sub domains
(.) in domain =1 → Legitimate(1)
(.) in domain =2 → Suspicious(0)
Else → Phishing (-1)

Age of certificate and issuer
HTTPS and age of certificate ≥ 1 Year → Legitimate (1)
HTTPS and issuer is not valid → Suspicious (0)
Else → Phishing (-1)

Domain registration length
Domains expires ≤1 year → Phishing (-1)
Else → Legitimate (1)

Favicon
Favicon loaded from external → Phishing (-1)
Else → Legitimate (1)

Using non-standard port
Port # is of the preferred status → Phishing (-1)
Else → Legitimate (1)

HTTPS in the domain
HTTPS in domain → Phishing (-1)
Else → Legitimate (1)

Request URL
% of request URL<22% → Legitimate (1)
%of request URL≥22% &<=61% → Suspicious(0)
Else → Phishing (-1)

URL of Anchor
% of URL of anchor <31% → Legitimate(1)
% of URL of anchor≥31% &≤67% → Suspicious(0)
Else → Phishing (-1)

Links in <Meta>, <Script> and <Link> tags
% of Links in meta,script&link<17% Legitimate
% of Links in meta,script&link≥17%&≤81% → Suspicious(0)
Else → Phishing (-1)

Server Form Handler
SFH is empty → Phishing (-1)
SFH refers to a different domain → Suspicious (0)
Else → Legitimate (1)

Submitting Information to Email
Using mail () or mailto: → Phishing (-1)
Else → Legitimate (1)

Abnormal URL
Host name not included → Phishing (-1)
Else → Legitimate (1)

Website Forwarding
# of redirect page≤1 → Legitimate (1)
# of redirect page≥2 & <4 → Suspicious(0)
Else → Phishing(-1)

Status Bar Customization
onMouseOver changes status bar → Phishing (-1)
Doesn't change status bar → Legitimate (1)

Disabling Right Click
Right click disabled → Phishing (-1)
Else → Legitimate (1)

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 6 June 2022, pp: 226-232 www.ijaem.net ISSN: 2395-5252

Using Pop-up Window
Popup window contains text fields → Phishing(-1)
Else → Legitimate(1)
IFrame Redirection
Using iframe → Phishing(-1)
Else → Legitimate(1)
Age of Domain
Age of domain ≥ 6 months → Legitimate(1)
Else → Phishing (-1)
DNS Record
No DNS record for the domain →Phishing(-1)
Else → Legitimate (1)
Website Traffic
Website rank<100,000 → Legitimate (1)
Website rank>100,000 →Suspicious (0)
Else → Phishing (-1)
PageRank
Page rank < 0.2 → Phishing (-1)
Else → Legitimate (1)
Google Index
Google Index → Legitimate (1)
Else → Phishing(-1)
Number of links pointing to page
# of link Pointing to webpage=0 →Phishing(-1)
#of link pointing to webpage>0&≤2→Suspicious(0)
Else → Legitimate(1)
Statistical-reports based feature
Host belongs to top phishing IPs → Phishing (-1)
Else → Legitimate (1)

Dataset is partitioned into training dataset and testing dataset. Finally, this resulting data split into 80% of dataset into train data and 20% of dataset into test data, which was further passed to machine learning model.

IV. METHODS AND ALGORITHMS

Support Vector Machine (SVM)

Support Vector Machine is also known as SVM. SVMs are a type of machine learning methodology that is built on the notion of structural risk reduction. In the realm of pattern recognition, it has a wide range of applications. For the estimation of the decision function, SVM generates a linear model basing on the support vectors. SVM identifies the best hyper plane that set apart the data without mistakes if the training data is linearly separable.

To change input patterns into higher dimensional feature space, SVM employs a nonlinear mapping technique. A linear SVM is used to categorize data sets that are linearly separable. The support vectors are the greatest design on the margins. The support vectors are all near to the separation hyper plane and are altered training patterns. The most challenging patterns to recognize are support vectors, which are training samples that make up the ideal hyper plane.

They're the most informative patterns in the categorization task, informally. The kernel function builds inner products in the input space to form machines. SVMs are one of the most robust prediction methods, being based on statistical learning frameworks or VC theory proposed by Vapnik (1982, 1995) and Chervonenkis (1974).

Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non- probabilistic binary linear classifier (although methods such as platt scaling exist to use SVM in a probabilistic classification setting).

SVM maps training examples to points in space so as to maximize the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

K-Nearest Neighbor (KNN)

K-Nearest Neighbor is also known as KNN. KNN algorithm is very easy and very efficient. The output data is calculated while the class with the peak frequency as of the k-most related instances. For each instance the votes for their class and the class with a large amount of votes is taken as prediction. It is used for classification and regression.

K-Nearest Neighbor Classifier

K Nearest Neighbors is a well-known supervised learning technique for classification that is straightforward to implement.The basic premise of KNN is that comparable goods are found close together, or that goods with comparable characteristics exist nearby. The KNN classifier uses mathematics to encapsulate the idea of object similarity, such as calculating distance between them.In KNN, the test sample is assigned a class value to the class of the majority of its nearest neighbors.The KNN algorithm is based on the K value, which determines the number of training neighbors to which a test sample is compared. Here we took K value as 5.

KNN Features

- KNN is a Supervised Learning algorithm that uses labeled input data set to predict the output of the data points.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 6 June 2022, pp: 226-232 www.ijaem.net ISSN: 2395-5252

- It is one of the most simple Machine learning algorithms and it can be easily implemented for a varied set of problems.
- It is mainly based on feature similarity. KNN checks how similar a data point is to its neighbor and classifies the data point into the class it is most similar to.

Decision Tree algorithm

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
- Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
- Step-3: Divide the S into subsets that contains possible values for the best attributes.
- Step-4: Generate the decision tree node, which contains the best attribute.
- Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Attribute Selection Measures

While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as Attribute selection measure or ASM. By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

- Information Gain
- Gini Index

Gradient Boosting

Gradient Boosting is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessor's error. In contrast to Adaboost, the weights of the training instances are not tweaked, instead, each predictor is trained using the residual errors of predecessor as labels. It is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

There is a technique called the Gradient Boosted Trees whose base learner is CART (Classification and Regression Trees). There is an important parameter used in this technique known as Shrinkage. Shrinkage refers to the fact that the prediction of each tree in the ensemble is shrunk after it is multiplied by the learning rate (eta) which ranges between 0 to 1. There is a trade-off between eta and number of estimators, decreasing learning rate needs to be compensated with increasing estimators in order to reach certain model performance. Since all trees are trained now, predictions can be made.

Each tree predicts a label and final prediction is given by the formula,

$y(pred) = y1 + (eta* r1)+(eta*r2) + ... + (eta * rN)$

The class of the gradient boosting regression in scikit-learn is Gradient Boosting Regressor. A similar algorithm is used for classification known as Gradient Boosting Classifier.Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting.The Gradient Boosting Classifier depends on a loss function. A custom loss function can be used, and many standardized loss functions are supported by gradient boosting classifiers, but the loss function has to be differentiable.Classification algorithms frequently use logarithmic loss, while regression algorithms can use squared errors. Gradient boosting systems don't have to derive a new loss function every time the boosting algorithm is added, rather any differentiable loss function can be applied to the system.

Gradient boosting systems have two other necessary parts: a weak learner and an additive component. Gradient boosting systems use decision trees as their weak learners. Regression trees are used for the weak learners, and these regression trees output real values. Because the outputs are real values, as new learners are added into the model the output of the regression trees can be

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 6 June 2022, pp: 226-232 www.ijaem.net ISSN: 2395-5252

added together to correct for errors in the predictions.

The main idea behind this algorithm is to build models sequentially and these subsequent models try to reduce the errors of the previous model. But how do we do that? How do we reduce the error? This is done by building a new model on the errors or residuals of the previous model. we know that the errors in machine learning algorithms are broadly classified into two categories i.e. Bias Error and Variance Error. As gradient boosting is one of the boosting algorithms it is used to minimize bias error of the model.

When the target column is continuous, we use Gradient Boosting Regressor whereas when it is a classification problem, we use Gradient Boosting Classifier. The only difference between the two is the "Loss function". The objective here is to minimize this loss function by adding weak learners using gradient descent. Since it is based on loss function hence for regression problems, we'll have different loss functions like Mean squared error (MSE) and for classification, we will have different for e.g log-likelihood.

Step-1 The first step in gradient boosting is to build a base model to predict the observations in the training dataset. For simplicity we take an average of the target column and assume that to be the predicted value

Step-2 The next step is to calculate the pseudo residuals which are (observed value – predicted value)

Step-3 we will build a model on these pseudo residuals and make predictions.

Step-4 In this step we find the output values for each leaf of our decision tree. That means there might be a case where 1 leaf gets more than 1 residual, hence we need to find the final output of all the leaves. TO find the output we can simply take the average of all the numbers in a leaf, doesn't matter if there is only 1 number or more than 1.

In gradient boosting decision trees, we combine many weak learners to come up with one strong learner. The weak learners here are the individual decision trees. All the trees are connected in series and each tree tries to minimise the error of the previous tree. Due to this sequential connection, boosting algorithms are usually slow to learn, but also highly accurate. In statistical learning, models that learn slowly perform better.

The weak learners are fit in such a way that each new learner fits into the residuals of the previous step so as the model improves. The final model aggregates the result of each step and thus a strong learner is achieved. A loss function is used to detect the residuals. For instance, mean squared error (MSE) can be used for a regression task and logarithmic loss (log loss) can be used for classification tasks. It is worth noting that existing trees in the model do not change when a new tree is added. The added decision tree fits the residuals from the current model.

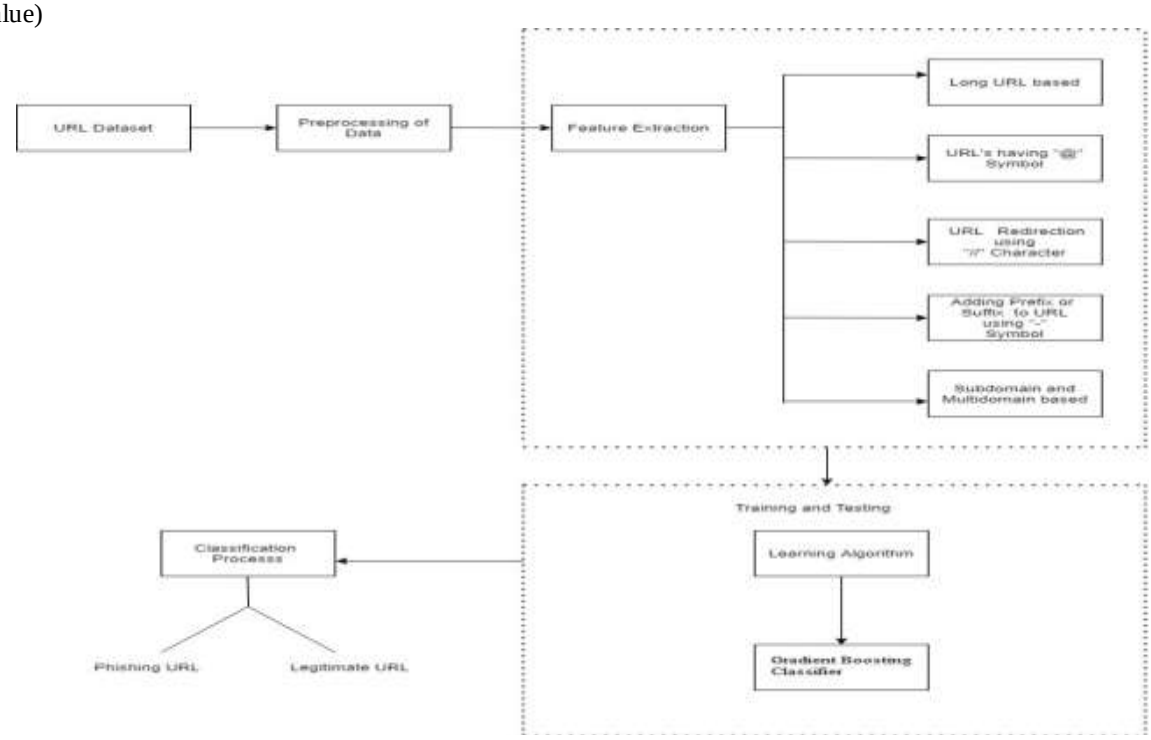


Figure 1 - BLOCK DIAGRAM

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 6 June 2022, pp: 226-232 www.ijaem.net ISSN: 2395-5252

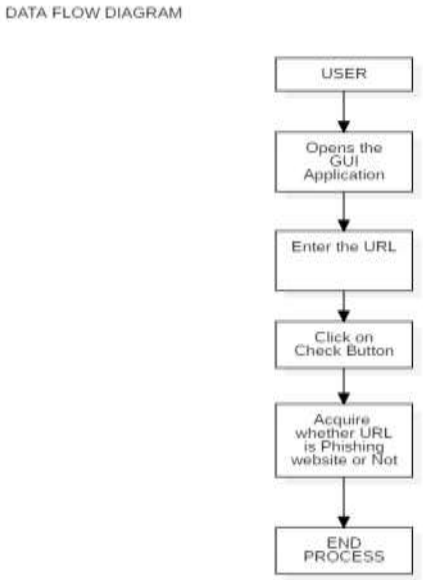


Figure2 - DATAFLOW DIAGRAM

V. PERFORMANCE METRICS

There are various metrics which we can use to evaluate the performance of ML algorithms, classification as well as regression algorithms. We must carefully choose the metrics for evaluating ML performance because −

- How the performance of ML algorithms is measured and compared will be dependent entirely on the metric you choose.
- How you weight the importance of various characteristics in the result will be influenced completely by the metric you choose.
- True Positive (TP) =Observation is positive, and is predicted to be positive.
- False Negative (FN)=Observation is positive, but is predicted negative.
- True Negative (TN)=Observation is negative, and is predicted to be negative.
- False Positive (FP)=Observation is negative, but is predicted positive.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$F1 \text{ score} = 2 * (precision * recall) / (precision + recall)$$

Classification Report of Gradient Boosting

	precision	recall	f1-score	support
-1	0.99	0.96	0.97	976
1	0.97	0.99	0.98	1235
accuracy			0.97	2211

We used different python libraries to implement the code and we performed exploratory data analysis(EDA) and pre-processing of dataset and we build the machine learning model using gradient boosting.

VI. RESULTS

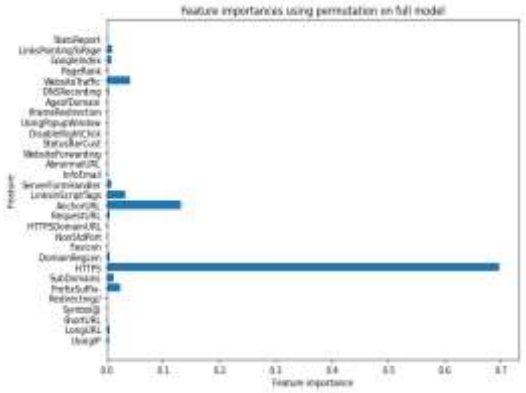
	ML Model	Accuracy	f1_score	Recall	Precision
0	Gradient Boosting Classifier	0.974	0.977	0.994	0.986
1	CatBoost Classifier	0.972	0.975	0.994	0.989
2	XGBoost Classifier	0.969	0.973	0.993	0.984
3	Multi-layer Perceptron	0.969	0.973	0.995	0.981
4	Random Forest	0.967	0.971	0.993	0.990
5	Support Vector Machine	0.964	0.968	0.980	0.965
6	Decision Tree	0.960	0.964	0.991	0.993
7	K-Nearest Neighbors	0.956	0.961	0.991	0.989
8	Logistic Regression	0.934	0.941	0.943	0.927
9	Naive Bayes Classifier	0.605	0.454	0.292	0.997

If we see the results, Gradient boosting classifier achieved the highest accuracy with 97.4% among all other algorithms and followed by CatBoost classifier with 97.2% and XGBoost Classifier with 96.9%.So, in the final

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 6 June 2022, pp: 226-232 www.ijaem.net ISSN: 2395-5252

implementation we implemented gradient boosting as it has the highest accuracy with 97.4%.

Feature Importance



VII. FURTHER ENHANCEMENT

We can further extend this Phishing website detection project to directly tie up with the cyber security department and other private white hat hackers to create a blacklist details of the phished websites and as a coin has both sides, As the technology is updating day by day at a very fast pace, the hackers or phishers also update themselves, invent new methods to avoid detection and also to fraud the general public who are not aware of the cyber security. We can also use artificial intelligence in this to identify and alert the people. More work on curating high- quality datasets could be performed. This could involve working with and reducing the size of pre-existing datasets, combining existing datasets together. Overall, phishing website detection remains an interesting and worthwhile challenge for academic institutions, businesses, cyber security experts, researchers etc and there is plenty of room for further study and analysis**.**

VIII. CONCLUSION

The final take away from this study is to explore various machine learning models, perform Exploratory Data Analysis on phishing dataset and understanding their features. Creating this notebook helped us to learn a lot about the features affecting the models to detect whether URL/website is safe or not, also we came to know how to tune a model and how they affect the model performance. The final conclusion on the Phishing dataset is that, some features like "HTTPS", "AnchorURL", "Website Traffic" have more importance to classify whether the website phished website or not. Gradient Boosting Classifier classified up to accuracy of 97.4% with respect to other classes and hence reduces the chance of becoming a victim of phishing attack.

REFERENCES

[1]. Rami Mustafa A Mohammad, T. L. Mccluskey, Fadi Thabtah 2015,"Intelligent Rule based Phishing Websites Classification
[2]. Samuel Marchal, Jerome Francois, Radu State, Thomas Engel 2017," PhishStorm: Detecting Phishing With Streaming Analytics"
[3]. https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15
[4]. https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners
[5]. https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/
[6]. https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_algorithms_performance_metrics.htm