Phishing Website Detection using Data mining and comparative study of Classifiers

G Thippa Reddy, MOHIT KEWALRAMANI, SAI SRIVASTAV RAJPUT, NISHANT KUMAR RAI
[1]thippareddy.g@vit.c.in,[2]mohit.kewalramani2015@vit.ac.in,[3]saisrivastav.rajput2015@vit.ac.in , [4]nishantkumar.rai2015@vit.ac.in

Abstract – Phishing sites are fake sites that are made by deceptive persons which are copy of genuine sites. These websites look like an official website of any company such as bank, institute, etc. The main aim of phishing is that to steal sensitive information of user such as password, username, pin number, etc. Victims of phishing attacks may uncover their money related delicate data to the attackers who may utilize this data for budgetary and criminal exercises. Different technical and non-technical approaches have been proposed to identify phishing sites. Non-Technical approach has no solution against the fast disappearance feature of phishing websites.

Data mining technique, one of the classifications of technical approach, has shown promising results in detection of phishing websites. As compared to non-technical approaches, data mining techniques can generate classification models which can make prediction on phishing websites in real-time.

I. INTRODUCTION

Year 1991 is known for the revolutionary change in internet usage [1]. In 1991, when internet was made available for commercial usage, many businesses were shifted to websites. Ecommerce grew with this huge change in business trends. This change provided the opportunity of Electronic data interchange and Electronic fund transfer. But with the trend of ecommerce, a new trend also came into the picture know as Cybercrimes. Cybercrime is any illegal behavior, directed by means of electronic operations, that targets the security of computer systems and the data processed by them [2].

Cyber Crime investigation cell of Mumbai, India [3] has categorized cybercrimes in following categories: Hacking, Child Pornography, Cyber stalking, Denial of service attack, Virus Dissemination, Software privacy, Internet Relay Chat(IRC) crime, Credit card fraud, Net extortion, Phishing and Internet fraud. According to George K. Kostopoulos in his book Cyberspace and Cyber security [4], vulnerabilities are introduced while system are being upgraded or adapted to new operational environment. People are becoming more 'tech savvy' with the improvement of technology. Internet has become an efficient network among people for communication and cooperation. Internet facilities can be used with personal computers, laptops, mobile phones, tablets, media player, e-reader etc. Through these digital devices linked by the internet, hackers also attack personal privacy using a variety of internet threats, such as, viruses, Trojans, Worms, botnet attacks, rootkits, adware, spam and social engineering platform. Phishing is such social engineering attack which attempts to obtain peoples' personal information by deception. Cyber criminals indulge in the practice of phishing are known as 'Phishers'. Phishing sites are manufactured by malicious individuals to show up as genuine sites. In general terms, Phishing is an act of sending an email to a client dishonestly asserting

to be an authentic business foundation trying to trick or trap the client to surrender private data that will be utilized for fraud. The effect is the rupture of data security through the bargain of private information and the casualty might finally suffer loss of money and other kinds. "Phishing" at first developed in 1990s [5]. The early programmers regularly utilize "ph" in place of "f" to create new words in the programmer's group, since they typically hack by telephones. Phishing is another word delivered from 'fishing'. The most of the time utilized in the phishing attack is to send messages to potential targets, which appeared to be sent by banks, online associations, or ISPs. In these messages, phishers will make up a few reasons, e.g. the secret key of user's credit card has been entered incorrectly or an offer of overhauling administrations in just one click. These messages will take the user to the self-made website page that demands the user to provide or change their account number and password through the hyperlink given in the e-mail. If the user submits the account number and secret key, the phishers then effectively gather the data at the server side, and can perform malicious activity with that data (e.g., pull back cash out from your record, changing of password etc.).

II. ANTI-PHIHSING TECHNIQUES

Detecting phishing websites is not an easy job to perform. It requires a lot of efforts to build a smart solution for detecting phishing websites. There are several anti-phishing techniques that have been evolved to protect our personal information against phishing attacks. All these techniques can be categorized into non-technical approaches and technical approaches [6].

A. Non-technical Approach

In non-technical approach involvement of humans are important. It can further be classified as legal solutions and training people.

1) **Legal solutions**: It is one of the non-technical methods to stop phishing activities. Legal solutions are followed by many countries. United States was first to enact laws against phishing activities. Phishing has been added to computer crime list in 2004 by Federal Trade Commission (FTC) which is an agency under U.S government that aims to promote internet users' protection. In the years 2005 and 2006, both the Australian and UK governments tighten its legal range against fraud by preventing the development of phishing websites and those who develop such websites can be given jail penalties. At-last, the Australian government also signed a partnership with Microsoft to teach the law enforcement officials how to check different cyber-crimes.

Nevertheless, legal solutions do not completely control phishing attacks since it is very difficult to trace them due to their fast disappearances in the cyber world.

2) **Training people**: Another method is educating a consumer in order to raise awareness about online crime. If internet-users could be convinced to always check the security indicators within the website the problem can be minimized. However, the important advantage for phishers to successfully cheat internet-users is that the majority of them lack basic knowledge of current online scams that may target them. Although raising awareness as much as possible about phishing to users may be seen as a promising direction it is still a tough task to implement.

B. Technical Approach

Typically, the two most important technical approaches to tackle the phishing attacks are the blacklist and the heuristic-based [7]. The two approaches are explained below:

1) **Blacklist Approach**: In the blacklist method, the requested URL is compared with a predefined set of phishing URLs. Most of the browsers use this approach. The drawback of this method is that it typically doesn't deal with all phishing websites since newly launched fake website takes a large amount of time before it is being added to the blacklist.

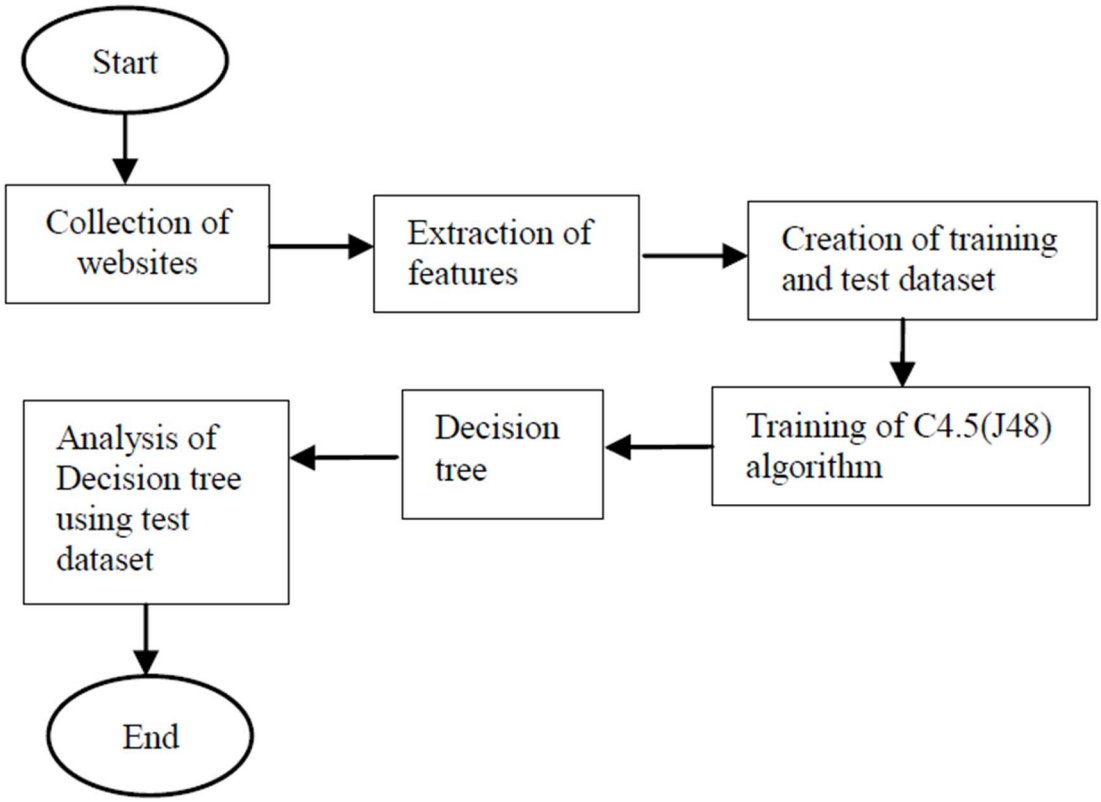
2) **Heuristic Approach**: In contrast to the blacklist method, the heuristic-based method can identify newly created phishing websites in real-time [8]. Several solutions using heuristic approach are present these days to handle phishing such as PhishZoo[9]. Moreover, some non-profit organizations such as APWG [10] and PhishTank[11] provide a platform where opinions as well as distribution of the best practices against phishing from users' experiences are stored. The success of an anti-phishing technique mainly depends on recognizing phishing websites within an acceptable time period.

III. WORKFLOW

The whole work is divided into several steps. Waikato Environment for Knowledge Analysis (WEKA) tool has been used for the analysis of classifier algorithms.

The different steps involved for the analysis of classifier algorithms are as follows:

- Collection of phishing websites from PhishTank.
- Extraction of features of phishing websites.
- Creation of training dataset and testing dataset on the basis of features extracted.
- Training of classifiers using training dataset and generation of decision tree classifier model.
- Prediction of missing values in testing dataset using classifier model.
- Evaluation of classifier model on the basis of different parameters.

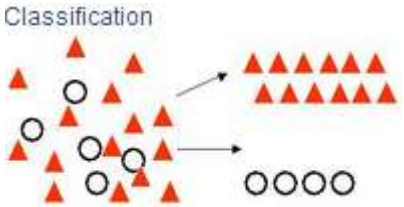


V. ALGORITHMS DESCRIPTION

1. Decision Tree CART Algorithm

The CART or Classification & Regression Trees methodology was introduced in 1984 by Leo Breiman, Jerome Friedman, Richard Olshen and Charles Stone as an umbrella term to refer to the following types of decision trees:

Classification Trees: where the target variable is categorical and the tree is used to identify the "class" within which a target variable would likely fall into.



Regression Trees: where the target variable is continuous and tree is used to predict its value.

The main elements of CART (and any decision tree algorithm) are:

1. Rules for splitting data at a node based on the value of one variable;
2. Stopping rules for deciding when a branch is terminal and can be split no more; and
3. Finally, a prediction for the target variable in each terminal node.

Some useful features and advantages of CART are:

- CART is nonparametric and therefore does not rely on data belonging to a particular type of distribution.
- CART is not significantly impacted by outliers in the input variables.
- You can relax stopping rules to "overgrow" decision trees and then prune back the tree to the optimal size. This approach minimizes the probability that important structure in the data set will be overlooked by stopping too soon.
- CART incorporates both testing with a test data set and cross-validation to assess the goodness of fit more accurately.
- CART can use the same variables more than once in different parts of the tree. This capability can uncover complex interdependencies between sets of variables.
- CART can be used in conjunction with other prediction methods to select the input set of variables.

2. Decision Tree J48 Algorithm

Decision tree J48 is the implementation of algorithm ID3 (Iterative Dichotomiser 3) developed by the WEKA project team. R includes this nice work into package RWeka.

The J48 Decision tree classifier follows the following simple algorithm. In order to classify a new item, it first needs to create a decision tree based on the attribute values of the available training data. So, whenever it encounters a set of items (training set) it identifies the attribute that discriminates the various instances most clearly. This feature that is able to tell us most about the data instances so that we can classify them the best is said to have the highest information gain. Now, among the possible values of this feature, if there is any value for which there is no ambiguity, that is, for which the data instances falling within its category have the same value for the target variable, then we terminate that branch and assign to it the target value that we have obtained.

For the other cases, we then look for another attribute that gives us the highest information gain. Hence we continue in this manner until we either get a clear decision of what combination of attributes gives us a particular target value, or we run out of attributes. In the event that we

run out of attributes, or if we cannot get an unambiguous result from the available information, we assign this branch a target value that the majority of the items under this branch possess.

Now that we have the decision tree, we follow the order of attribute selection as we have obtained for the tree. By checking all the respective attributes and their values with those seen in the decision tree model, we can assign or predict the target value of this new instance. The above description will be more clear and easier to understand with the help of an example.

3. Naïve Bayes Algorithm

The Naïve Bayes classifier works on a simple, but comparatively intuitive concept. Also, in some cases it is also seen that Naïve Bayes outperforms many other comparatively complex algorithms. It makes use of the variables contained in the data sample, by observing them individually, independent of each other.

When

$X = \langle X_1, X_2, \cdots, X_n \rangle$, X_i : discrete or continuous , Y : discrete

Naive Bayes classifier

$$P(Y = y_k | X_1 \ldots X_n) \overset{\text{Bayes}}{=} \frac{P(Y = y_k) P(X_1 \ldots X_n | Y = y_k)}{\sum_j P(Y = y_j) P(X_1 \ldots X_n | Y = y_j)}$$

Conditional Independence

$$\overset{\text{Naive Bayes}}{=} \frac{P(Y = y_k) \prod_i P(X_i | Y = y_k)}{\sum_j P(Y = y_j) \prod_i P(X_i | Y = y_j)}$$

Then, the most probable value of Y (Answer) is

$$Y \leftarrow \arg\max_{y_k} \frac{P(Y = y_k) \prod_i P(X_i | Y = y_k)}{\sum_j P(Y = y_j) \prod_i P(X_i | Y = y_j)}$$

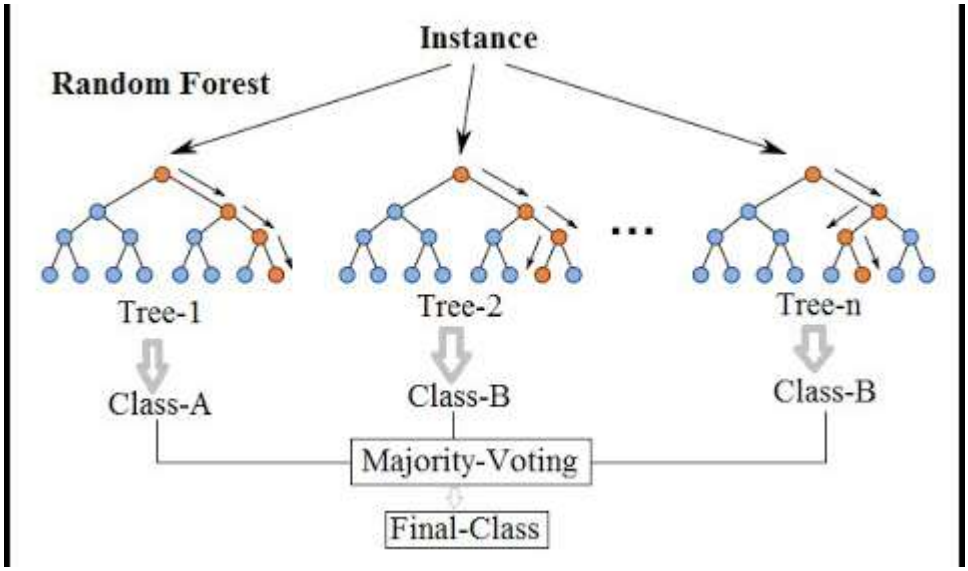
Since the denominator does not depend on y_k, simply

$$Y \leftarrow \arg\max_{y_k} \underbrace{P(Y = y_k)}_{\text{② prior}} \prod_i \underbrace{P(X_i | Y = y_k)}_{\text{① likelihood}}$$

The Naïve Bayes classifier is based on the Bayes rule of conditional probability. It makes use of all the attributes contained in the data, and analyses them individually as though they are equally important and independent of each other. For example, consider that the training data consists of various animals (say elephants, monkeys and giraffes), and our classifier has to classify any new instance that it encounters. We know that elephants have attributes like they have a trunk, huge tusks, a short tail, are extremely big, etc. Monkeys are short in size, jump around a lot, and can climb trees; whereas giraffes are tall, have a long neck and short ears.

The Naïve Bayes classifier will consider each of these attributes separately when classifying a new instance.

4. Random Forest Algorithm



5. Multilayer perceptron – Back propagation Algorithm

Input: Data set D, learning rate l, network **Output:** Trained Neural Network

(1) Initialize all weights and biases in *network*;
(2) while terminating condition is not satisfied {
(3) for each training tuple X in D {
(4) // Propagate the inputs forward:
(5) for each input layer unit j {
(6) $O_j = I_j$; // output of an input unit is its actual input value
(7) for each hidden or output layer unit j {
(8) $I_j = \sum_i w_{ij} O_i + \theta_j$; //compute the net input of unit j with respect to the previous layer, i
(9) $O_j = \frac{1}{1+e^{-I_j}}$; } // compute the output of each unit j
(10) // Backpropagate the errors:
(11) for each unit j in the output layer
(12) $Err_j = O_j(1-O_j)(T_j - O_j)$; // compute the error
(13) for each unit j in the hidden layers, from the last to the first hidden layer
(14) $Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$; // compute the error with respect to the next higher layer, k
(15) for each weight w_{ij} in *network* {
(16) $\Delta w_{ij} = (l) Err_j O_i$; // weight increment
(17) $w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update
(18) for each bias θ_j in *network* {
(19) $\Delta\theta_j = (l) Err_j$; // bias increment
(20) $\theta_j = \theta_j + \Delta\theta_j$; } // bias update
(21) } }

IV. DATASET USED

Link - https://archive.ics.uci.edu/ml/datasets/phishing+websites

Data Set Information:

One of the challenges faced by our research was the unavailability of reliable training datasets. In fact this challenge faces any researcher in the field. However, although plenty of articles about predicting phishing websites have been disseminated these days, no reliable training dataset has been published publically, may be because there is no agreement in literature on

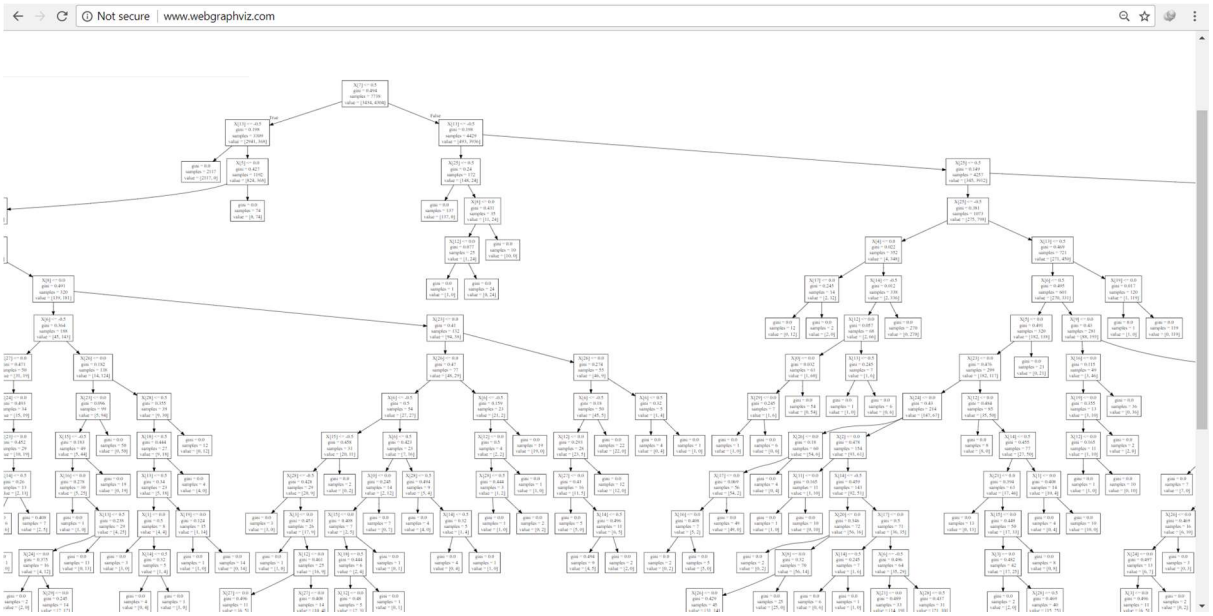
the definitive features that characterize phishing webpages, hence it is difficult to shape a dataset that covers all possible features.
In this dataset, there are 30 attributes and the 31st attribute corresponds to the result attribute and we shed light on the important features that have proved to be sound and effective in predicting phishing websites. In addition, we propose some new features.

```
'data.frame':   2456 obs. of  31 variables:
 $ has_ip               : Factor w/ 2 levels "0","1": 2 1 1 ...
 $ long_url             : Factor w/ 3 levels "0","1","-1": 2 2 1 ...
 $ short_service        : Factor w/ 2 levels "0","1": 1 1 1 ...
 $ has_at               : Factor w/ 2 levels "0","1": 1 1 1 ...
 $ double_slash_redirect: Factor w/ 2 levels "0","1": 2 1 1 ...
 $ pref_suf             : Factor w/ 3 levels "0","1","-1": 3 3 3 ...
 $ has_sub_domain       : Factor w/ 3 levels "0","1","-1": 3 1 3 ...
 $ ssl_state            : Factor w/ 3 levels "0","1","-1": 3 2 3 ...
 $ long_domain          : Factor w/ 3 levels "0","1","-1": 1 1 1 ...
 $ favicon              : Factor w/ 2 levels "0","1": 1 1 1 ...
 $ port                 : Factor w/ 2 levels "0","1": 1 1 1 ...
 $ https_token          : Factor w/ 2 levels "0","1": 2 2 2 ...
 $ req_url              : Factor w/ 2 levels "1","-1": 1 1 1 ...
 $ url_of_anchor        : Factor w/ 3 levels "0","1","-1": 3 1 1 ...
 $ tag_links            : Factor w/ 3 levels "0","1","-1": 2 3 3 ...
 $ SFH                  : Factor w/ 2 levels "1","-1": 2 2 2 ...
 $ submit_to_email      : Factor w/ 2 levels "0","1": 2 1 2 ...
 $ abnormal_url         : Factor w/ 2 levels "0","1": 2 1 2 ...
 $ redirect             : Factor w/ 2 levels "0","1": 1 1 1 ...
 $ mouseover            : Factor w/ 2 levels "0","1": 1 1 1 ...
 $ right_click          : Factor w/ 2 levels "0","1": 1 1 1 ...
 $ popup                : Factor w/ 2 levels "0","1": 1 1 1 ...
 $ iframe               : Factor w/ 2 levels "0","1": 1 1 1 ...
 $ domain_Age           : Factor w/ 3 levels "0","1","-1": 3 3 1 ...
 $ dns_record           : Factor w/ 2 levels "0","1": 2 2 2 ...
 $ traffic              : Factor w/ 3 levels "0","1","-1": 3 1 2 ...
 $ page_rank            : Factor w/ 3 levels "0","1","-1": 3 3 3 ...
 $ google_index         : Factor w/ 2 levels "0","1": 1 1 1 ...
 $ links_to_page        : Factor w/ 3 levels "0","1","-1": 2 2 1 ...
 $ stats_report         : Factor w/ 2 levels "0","1": 2 1 2 ...
 $ target               : Factor w/ 2 levels "1","-1": 1 1 1 ...
```

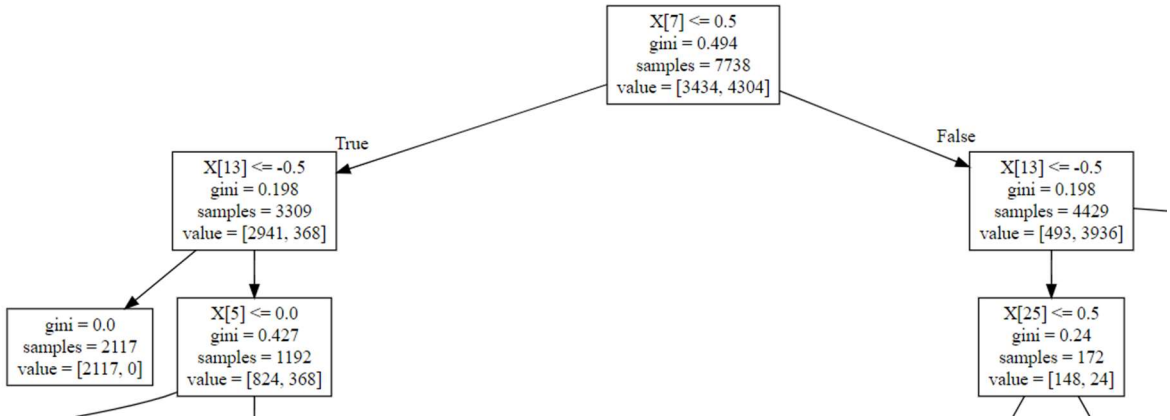
VI. RESULTS

1. **Running Decision tree classifier on the dataset using python's sklearn library. It uses CART Algorithm. On running the code, it produced accuracy of 88%**

Output –



Each node in the decision tree output contains the attribute number, gini value and other information. The output of the python code gets stored in tree.dot file. We use the saved information on www.webgraphviz.com to print the decision tree.



Code snippet –

```
from sklearn import tree
from sklearn.metrics import accuracy_score
import numpy as np
import graphviz
def load_data():
    training_data = np.genfromtxt('dataset.csv', delimiter=',', dtype=np.int32)
    inputs = training_data[:,:-1]
    outputs = training_data[:, -1]
```

```
    training_inputs = inputs[:7738]
    training_outputs = outputs[:7738]
    testing_inputs = inputs[7739:]
    testing_outputs = outputs[7739:]
    return training_inputs, training_outputs, testing_inputs, testing_outputs, training_data
if __name__ == '__main__':
    print("Tutorial: Training a decision tree to detect phishing websites")
    train_inputs, train_outputs, test_inputs, test_outputs, training_data = load_data()
    print ("Training data loaded.")
    classifier = tree.DecisionTreeClassifier()
    print ("Decision tree classifier created.")
    print ("Beginning model training.")
    classifier = classifier.fit(train_inputs, train_outputs)
    print ("Model training completed.")
    predictions = classifier.predict(test_inputs)
    print ("Predictions on testing data computed.")
    dot_data = tree.export_graphviz(classifier, out_file="tree.dot")
    graph = graphviz.Source(dot_data)
    accuracy = 100.0 * accuracy_score(test_outputs, predictions)
```

2. **Naïve Bayes Algorithm using WEKA tool. Accuracy – 92.9806%**

```
=== Summary ===

Correctly Classified Instances       10279               92.9806 %
Incorrectly Classified Instances       776                7.0194 %
Kappa statistic                          0.8573
Mean absolute error                      0.0894
Root mean squared error                  0.2304
Relative absolute error                 18.1197 %
Root relative squared error             46.3892 %
Total Number of Instances            11055

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.904    0.050    0.936      0.904    0.919      0.858    0.981     0.979     -1
               0.950    0.096    0.926      0.950    0.938      0.858    0.981     0.984     1
Weighted Avg.  0.930    0.076    0.930      0.930    0.930      0.858    0.981     0.982

=== Confusion Matrix ===

    a    b   <-- classified as
 4427  471 |    a = -1
  305 5852 |    b = 1
```

3. Random Forest Algorithm using WEKA tool. Accuracy – 97.2863%

```
=== Summary ===

Correctly Classified Instances       10755               97.2863 %
Incorrectly Classified Instances       300                2.7137 %
Kappa statistic                          0.9449
Mean absolute error                      0.051
Root mean squared error                  0.1431
Relative absolute error                 10.3239 %
Root relative squared error             28.8082 %
Total Number of Instances            11055

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.961    0.017    0.978      0.961    0.969      0.945    0.996     0.995     -1
               0.983    0.039    0.969      0.983    0.976      0.945    0.996     0.995     1
Weighted Avg.  0.973    0.030    0.973      0.973    0.973      0.945    0.996     0.995

=== Confusion Matrix ===

    a    b   <-- classified as
 4705  193 |    a = -1
  107 6050 |    b = 1
```

4. Decision Tree (J48) Algorithm using WEKA tool. Accuracy – 95.8752%

```
=== Summary ===

Correctly Classified Instances       10599               95.8752 %
Incorrectly Classified Instances       456                4.1248 %
Kappa statistic                          0.9162
Mean absolute error                      0.0567
Root mean squared error                  0.1853
Relative absolute error                 11.4861 %
Root relative squared error             37.3035 %
Total Number of Instances            11055

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.942    0.028    0.964      0.942    0.953      0.916    0.984     0.983     -1
               0.972    0.058    0.955      0.972    0.963      0.916    0.984     0.978     1
Weighted Avg.  0.959    0.045    0.959      0.959    0.959      0.916    0.984     0.980

=== Confusion Matrix ===

    a    b   <-- classified as
 4615  283 |    a = -1
  173 5984 |    b = 1
```

5. Multilayer Perceptron, which uses BACKPROPAGATION algorithm for classification, using WEKA tool. Accuracy – 95.8752%

```
=== Summary ===

Correctly Classified Instances       10679               96.5988 %
Incorrectly Classified Instances       376                3.4012 %
Kappa statistic                          0.931
Mean absolute error                      0.0356
Root mean squared error                  0.1632
Relative absolute error                  7.2135 %
Root relative squared error             32.8553 %
Total Number of Instances            11055

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.957    0.027    0.966      0.957    0.961      0.931    0.994     0.993     -1
               0.973    0.043    0.966      0.973    0.970      0.931    0.994     0.995     1
Weighted Avg.  0.966    0.036    0.966      0.966    0.966      0.931    0.994     0.994

=== Confusion Matrix ===

    a    b   <-- classified as
 4687  211 |    a = -1
  165 5992 |    b = 1
```

VII. CONCLUSION

On running all the algorithms on Weka platform on the same dataset with Percentage Split of 70%, it was found that **Random Forest Algorithm** performed the best with a high accuracy score of **97.2863**%.

VIII. References

[1]. B. Leiner, V. Cerf, D. Clark, R. Kahn, L. Kleinrock, D. Lynch, J. Postel, L. Roberts and S. Wolff, "Internet Society," 2012. [Online].

Available: www.internetsociety.org/internet/what-internet/historyinternet/brief-history-internet

[2]. P. B. Pathak, "Cybercrime: A Global Threat to Cybercommunity," International Journal of Computer Science & Engineering Technology (IJCSET), vol. 7, no. 3, pp. 46-49, 2016.

[3]. "Cyber Crime Investigation Cell," 2005. [Online]. Available: www.cybercellmumbai.gov.in

[4]. G. Kostopoulos, Cyberspace and Cybersecurity, CRC Press, 2012.

[5]. M. Aburrous, M. A. Hossain, K. Dahal and F. Thabtah, "Associative classification techniques for predicting e-banking phishing websites," in MCIT'2010: International Conference on Multimedia Computing and Information Technology, 2010.

[6]. M. M. Al-Daeef, N. Basir and . M. M. Saudi, "A Review of Client- Side Toolbars as a User-Oriented Anti-Phishing Solution," Advanced Computer & Communication Engineering Technology, Lecture Notes in Electrical Engineering, pp. 427-437, 2016.

[7]. G. Aaron, "The State of Phishing," Computer Fraud {&} Security Bulletin, no. 6, pp. 5-8, 2010.

[8]. D. Miyamoto, H. Hazeyama and Y. Kadobayashi, "An Evaluation of Machine Learning-Based Methods for Detection of Phishing Sites," in International Conference on Neural Information Processing, 2008.

[9]. S. Afroz and R. Greenstadt, "PhishZoo: Detecting Phishing Websites by Looking at Them," in 2011 IEEE Fifth International Conference on Semantic Computing, 2011.

[10]. "APWG," 2011. [Online]. Available: www.antiphishing.org

[11]. "PhishTank," 2006. [Online]. Available: https://www.phishtank.com/