PHISHING WEBSITE DETECTION SYSTEM BASED ON ENHANCED iTREE CLASSIFIER

D. Muthu Krishnan[1] and V. Subramaniyaswamy[2]
[1]Advanced Computing-M.Tech, School of Computing, SASTRA University, India
[2]School of Computing, SASTRA University, India
E-Mail: muthu.jupitor@gmail.com

ABSTRACT

Nowadays, a lot of attacks have cropped up for phishing of emails and password of users. In order to get one's confidential information like passwords, bank details such as debit or credit card numbers illegally, phishing act can be done. For doing an identity theft or financial gain or some fraudulent activities, an individual or group of persons gets confidential information from unsuspecting victims via email. This study proposes data mining techniques to classify phishing data's and then attempts to rectify phishing. A decision classifier tree model is used for detecting phishing datasets and another hybrid session based model is also developed to escape from such phishing attacks. The use of the session based unique password is scrutinized to protect the privacy of the users which is not revealed by attacks or infected systems. The proposed model generates a novel minimum spanning tree called iTree which takes only a minimum time to construct it. The proposed iTree model classifies the inputs which are resistant to a number of phishing attacks and is also resilient to keyboard logging as well, thus ensuring reliability. The tree classifier model will be able to identify any types of attacks in the future. The proposed models of session based authentication and data mining based decision tree classification are much more accurate in the usage of the antiphishing than the existing method.

Keywords: iTree, anti phishing, data mining, classification, clustering, decision tree.

INTRODUCTION

"Phishing" means getting the sensitive private information like passwords and bank details directly or indirectly by masquerading like an entity which is a trustworthy one [6]. Phishing can be done by sending an attached receivable link in an email. In order to deceive the unsuspecting users, phishing can be done through social networking sites, online payment modules or auction sites. This type of fake emails contains the malware infected website link in it [15]. This website's visual similarity is as same as that of original websites because its look and feel is same as that of original one. Thus the users get fooled by believing it as an original website. Actually, in order to deceive the users and theft their identity, phishing can be used which is an example of social engineering technique [8]. The term "phishing" was coined in the year 1995 as from fishing meaning to snare or trap with bait [4]. If the malicious link like the bait is clicked or the malicious attachment is opened then, the unsuspecting user's information and passwords will be stolen.The proposed model presents a method for detecting such phishing attacks using machine learning techniques, i.e. data mining by extracting features. The phishing dataset is classified, rules are generated and decision trees of J48, C4.5 are formed by which predictions can be made. The second model is a decision tree model called iTree which visualizes the decision and rules where phishing techniques may be contained. In order to filter the phishing attacks, browser toolbars are used which is the former attempt, i.e. Spoofguard and Netcraft [10]. This toolbar holds 85% of accuracy when detecting the phishing websites [9]. But as compared to the email filtering, this toolbar holds both advantages and disadvantages in it. The main disadvantage of this toolbar is that, there is an amount of decrease in contextual information. An attack is delivered to the user by providing the context via email. The second model is email filter; it will entice the user to make an action based on the exact word. But the filter which operates in the browser from email client doesn't know the exact word. Header information was accessed by email filter model. It knows the information about who send the message and it also knows about the message routing process. Another main disadvantage of the toolbar is that it can't shield the user completely from the decision making process. [12] In order to fight against phishing attacks, two technical methods like heuristic based and blacklist methods are used. Blacklist method compares the predefined phishy URL with the requestedURL. But the main problem with this method is that it can't deal with all websites since the newly added phishy websites takes more time to get updated in blacklist [14]. Hence, the drawback in above mentioned solution tends to the new solution against phishing websites. The heuristic based method will diagnose the newly added fake websites in real time [2]. In order to handle phishing, McAfee offers some efficient solution and APWG also offers some solution against phishing, which is a non-profit organization [3]. Meanwhile PhishTank [5] and MillerSmiles [1] provided forums of opinions about phishing websites and fake URL's thus creating awareness among the peoples. By dismissing the risk of this warning and by hiding this risk from user, the proposed model can able to filter out the phishy emails before the user watches it. It prevents the productivity loss for those users who are suffering from it. These users consume more time to fetch process and discard the email attack. Here the browser with toolbar implementation doesn't have this particular content in it which is a drawback of the existing method.

PROPOSED WORK

The proposed model uses machine-learning based approach where phishing datasets are classified. The dataset is used to train the model and then rules are generated. Classifier takes input and the result is like to make a decision, whether the data in input were designed to defraud the user or not. Two classes are there during the classification of email. They are good emails and phishy emails. We identified that the feature collection is successful while diagnosing phishing. Then a tree model is generated using J48, iTree and Rep Tree. The proposed iTree classifier is resilient to all features and it selects the features. The instance classification problem of software failures arises because of two common situations.

a) To report the enormous amount of failures faced in deploying software via users,

b) By executing the synthetic test suite, enormous amount of failures will be induced.

Number of failures fell in the fewest number of groups in both cases; each one holds the failure which is caused by the same software defect. It is desirable to identify these groups before diagnosing the failure causes because to make the corrective maintenance as easy one. Because, it indicates the number of defects which is responsible for failures that how every defect frequently causes the failure, and which failures is relevant in order to diagnose specific defects. Sometimes, manifestingly and distinctively failure causes a particular defect. By observing the output of program, we can easily determine this failure is for the same reason. Further, we can't classify more other failures easily.As software systems continue to grow in size and complexity; they are increasingly designed to be configurable. This is desirable because it enables systems to be more portable, reusable, and extensible. At the same time configurability can greatly complicate software development tasks such as testing. Because each configuration can contain unique faults and therefore each configuration may need to undergo expensive testing which is something generally infeasible in practice.

The architecture of the proposed model is shown in Figure-1 as given below:

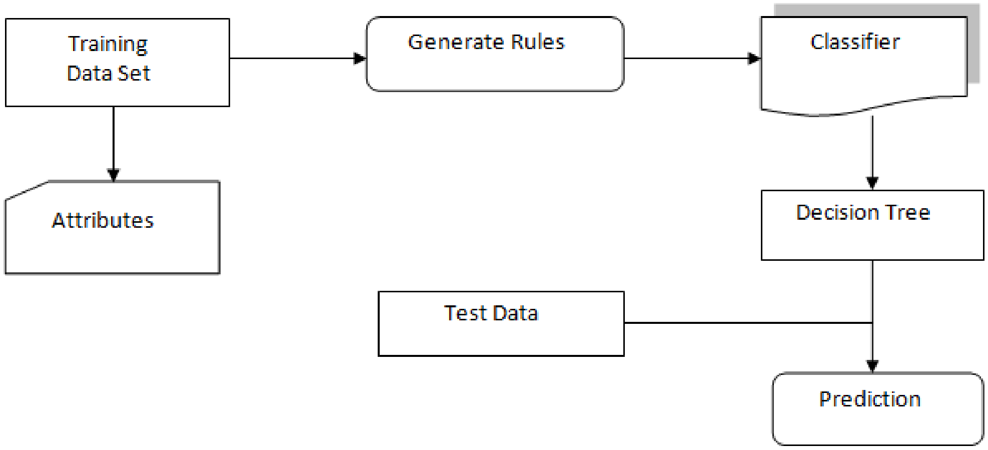


Figure-1. Architecture diagram of the decision tree classifier model.

PROBLEM IDENTIFICATION

By applying the multivariate data mining/data analysis to the execution profiles, we can simplify the classification problem of failure. To record and analyze the execution, this approach needs three types of information.

a) Reflection of failure in execution profiles,

b) Auditing information- Used to confirm reported failures and,

c) Diagnostic information- Used to determine the failure causes. If two failures have same causes then it is called same failure type in anti-phishing.

In order to prevent anti-phishing, tree configuration form like semi automated strategy holds multivariate visualization techniques and it also holds both supervised and unsupervised pattern classification techniques to execute profiles. We can classify the software failures using semi automated strategy. In addition to the failure of three subject programs, experimental measures suggest that, it is effective to group failures for the same cause. This strategy calls for anti-phishing's manual investigation to confirm or if necessary to refine the initial classification. The result indicates that certain groups in initial classification can be split or merge with other groups. However, it offers limited guidance, that's how the strategy splits the group or merges the group.

IMPLEMENTATION

The proposed phishing detection model is implemented as follows in the steps illustrated below:

First, a website user browses the internet or clicks the link in the email. This action will direct the user to legitimate or phishy website. Basically this website is the training data set. The training data set representation of our sample website is shown in Table-1.

Table-1. Training data set.

Number of instances	#1	#2	Class_Value
	x4	y2	z1
	x2	y1	z1
	x1	y1	z2
	x1	y2	z2
	x1	y1	z2
	x1	y2	z1
	x1	y2	z1
	x1	y1	z2
	x1	y3	z1
	x3	y1	z1

Where # represents the attribute and, Class_Value represents the class. This is basically done by writing a script embedded within the browser. Next the classifier, extract features from test data a stores it in the data tree model. Within browser, the proposed model will be active and based on the generated rules it guesses the type of the website type.Next phishing data will be collected from historical websites. Table-2 shows that of former historical data where candidate rules are generated here as given below.

Table-2. Candidate rules from data.

Item of rules		Support_Value (%)	Confidence_Value (%)
#1	Class_Value		
x1^y2	z1	20	66
x1^y1	z2	30	100
y2	z1	30	75
y1	z2, z1	25	50
x1	z2, z1	35	50

Test data are predicted by rules of generating classifier based on the extracted feature similarity. Table-3 shows that the iTree classifier is constructed from previous Table-2's data.

Table-3. iTree algorithm classifier.

Item of rules		**Support_Values** (%)	**Confidence_Values** (%)
#1	**Class_Value**		
x1	z2, z1	35	50.00
y1	z2, z1	25	50.00
y2	z1	30	75
x1^y1	z2	30	100

If the browsed website is a legitimate one, then don't take any action. But if it is phishy, then warn the user via proposed model. In order to generate the rules, the above step uses the strategy of machine classification learning. Here three important steps are there. That is, discovery of rules, building a classifier and final one is the assignment of class to test data. This algorithm iterates the training data set which contains phishing data by the way the rule is extracted and generated. By merging the same attribute and different classes resulting rules, multi-label rules are produced. Further the redundant rules without training data coverage are exterminated from the data set. The second step output is a decision tree classifier which holds multi and single-label rules. The last step is to examine the test data in order to ensure the classifier performance. To predict the website, we have to match the test data features with classifier's rules and states its class whether its phishy or not phishy. Hence the rules are produced and generated in the classifier. Let us assume that support and confidence has been set to 20% and 40% respectively. Here, the support and confidence value is a fixed one which is based on the formulae. The Support value is based on *suppthreshold <= suppcount(r)/*|D| formula, whereas the confidence value is based on *confthreshold <= suppcount(r)/actoccr(r)* formula.

Where D = number of instances,
suppcount(r) = support count of rule item r and,
actoccr(r) = an actual occurrence of rule item r.

The above table displays the candidate rules, training data set and, iTree algorithm classifier.

The algorithm first generates the rules, and then it checks for the candidate rule which is extracted from a similar body of the current rule. If the condition is true, then it appends the current rule with the extracted rule in order to produce the new rule. Two class labels are connected by attribute value (i.e.) true and false with different frequencies respectively. The normal Association Classification algorithms produce only one rule for this attribute value, while it simply discards the other classes. Thus more the number of occurrences in the training, better will be the upcoming results. Thus the proposed model enables the website users to have an additional knowledge. By this way, we get one more solution for phishing scenario.Secondly, due to the presence of multiple classes, the predictive accuracy will be improved. It is associated with another rule that has different weights based on the frequency of the training data set. Thus more rules are present whenever the test data classification for phishing. Hence, there will be more rules to classify it than the earlier methods and thus is accurate.In order to consider phishy, candidate rule must pass the measure of confidence and support. Our classifier will generate confidence and support after updating the actual frequency and confidence value. The candidate rules represent the possible candidate feature extracted multi-label rules. Thus, when this rule extraction is completed the classifier sorts all candidate rules based on the length of the rule, support, and confidence. Now, the candidate rule is ready for the evaluation of phishing training data set and also chooses the best one which makes the classifier so accurate.

DATA SET CLUSTERING

Data clustering technique (Vector Space Model) is relying on anti-phishing data set's single term analysis. An informative feature that contains more phrases and their weights are necessary to achieve data clustering. Applications like Data's automatic categorization, Data's taxonomy building and grouping the results of search engine benefit from data clustering since it is useful for these applications [11]. Hierarchical clustering method provides better improvement to achieve result. For successful hierarchical clustering, our project presents two key parts. Data index model is the first part. Data index graph not only rely on single-term indexes, but also allows special importance for the incremental construction efficiency of indexed data set. To judge the similarity between the two data, it provides the efficient phrase matching. If we can't choose for index phrases, then revert will happen in the vector space model's compact representation, because of the flexible nature of this model. iTree algorithm's incremental data clustering is the second part. It maximizes the cluster tightness by watching the pairwise distribution of data similarity in clusters. Both two phrases are based on Maximization expectation and Model of Gaussian Mixture [17]. Because of the accurate data similarity calculation and robustness, two component's combination creates an underlying model in it. It leads to improved results for web data clustering.

METHODOLOGY

MEASUREMENT OF DISTANCE

Selecting the distance measure is very important one. It determines about how the two element's similarity was calculated. Also the cluster's shape is influenced by it, based on the distance some elements will be close while some elements will far away. An example is two Dimensional space, based on the usual norms between the points (x=1, y=0) and the origin (x=0, y=0), the distance will be 1. But based on 1 or 2 or infinity norms between the points (x=1, y=1) and the origin, the distance will be 2, 1 or √2.

TO FIND THE DISTANCE

- 2-norm distance is nothing but Euclidean distance or crow flies. In the research of health psychology, most common distance is squared Euclidean or Euclidean distance while reviewing the cluster analysis.
- 1-norm distance is nothing but Taxicab distance or Manhattan distance.
- Both for variable's correlation and different scales, Mahalanobis distance corrects the data.
- In the process of clustering the high dimensional data, the angle between the two vectors is used as the distance to it. Also look at the inner space product.
- To change from one member to another, a small number of substitutions are needed. This small number of substitutions was measured by edit or Hamming distance.

CREATION OF CLUSTERS

The iTree model builds or breaks up the cluster hierarchy where the build is agglomerative and break up is divisive. iTree is the traditional representation of this hierarchy. At one end, single cluster holds an individual element and at the other end, single cluster holds every element with it. Algorithm of divisive begins from the tree root and the algorithm of agglomerative begins from the tree leaves. By cutting the tree at a given height, we can give clustering to selected precision. An example given below is, the cluster will be yielded by cutting after second row is {1} {2 3} {4 5} {6}. The cluster will be obtained by cutting after third row is {1} {2 3} {4 5 6}, this is called as coarser clustering. It has large clusters with small numbers with it.

AGGLOMERATION

For example, Distance metric is the Euclidean distance when we cluster this data.

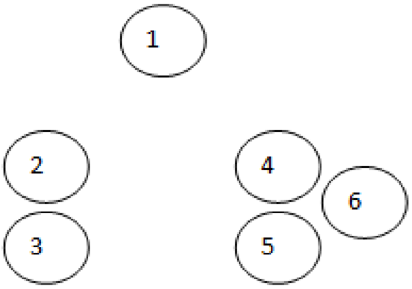


Figure-2. Unprocessed data.

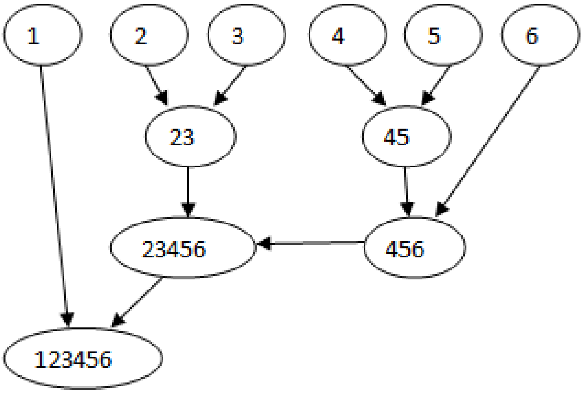


Figure-3. After process.

By merging the cluster, this method builds the hierarchy of individual elements. For example, here represented elements are six elements (i.e.) {1} {2} {3} {4} {5} and {6}. To determine whether which element have to be merged with cluster is the initial step. Based on the chosen distance, it takes the two closest elements usually.

One can construct the distance matrix in this stage where the k-th row to n-th column total numbers is same as that of k-th to n-th element's distance. Then, rows and columns merge the clusters and update the distance during the clustering progresses. To implement this type of clustering, this is the common way. In single-linkage clustering page, agglomerative algorithm is described which will easily adapt for linkage of different types of clusters.

Between the two clusters $\mathcal{A}$ and $\mathcal{B}$'s distance as follows:

- A linkage clustering is called complete-linkage clustering if the distance between every cluster's element are maximized

  max{ d(a, b) : a € $\mathcal{A}$, y € $\mathcal{B}$ }

- A linkage is called single-linkage clustering if the distance between every cluster's element are minimized
  min{ d(a, b) : a € $\mathcal{A}$, y € $\mathcal{B}$ }
- A linkage is called average- linkage clustering, if the distance between every cluster's element are mean distance.

$$\frac{1}{|\mathcal{A}|\cdot|\mathcal{B}|}\sum_{a\in\mathcal{A}}\sum_{b\in\mathcal{B}} d(a,b)$$

- All cluster variance (intra) is summed first.
- In order to make the cluster as merged, we should increase the variance (criterion of Ward).
- From same distribution function, the probability of candidate cluster's spawn called V-linkage function is obtained.

Current agglomeration's distance between the two clusters is greater than the former agglomeration's cluster distance and clustering will be stopped when the clusters are too far away, which is the distance criterion one, or when a little number of cluster presents which is number criterion one.

- Number of clusters has been chosen first.
- To be in the cluster, randomly assign to every point coefficients.
- Till the algorithm converges, repeat it.
- By using the above formula, compute the value of centriod for every cluster.
- By using the above formula, compute the point's coefficients for the point being in a cluster.

iTREE ALGORITHM

```
Upload an antiphishing dataset
Count the number of attributes
For each itemset in the data, assign a class
Iterate to the next point in the dataset
While
For this itemset class
Find the next item
If present then
One cluster is got
Mark it as a node in the root
Else
Load the next class item (for another cluster)
Check for distance
Traverse the tree and all its nodes
Till the last feature is extracted
Mark it as a child node of the root node
End if
Repeat for n classes
Do loop
Now check for the left out feature sets
Count the clusters
Draw the tree as a graph with node attributes
```

DATA SET PREPARTION AND LOADING

The system datasets of the various software's and considered architectures are first collated in this module. It is composed of two components which is connected in turn rid the data set of irrelevant features after removal and the process is called redundant feature elimination. The process will happen to extract the relevant features for target concept and also to eliminate irrelevant features. By choosing representative from distinct feature cluster, the latter removes the redundant features. Hence the final subset is produced by it. The model considers the first n features of the antiphishing dataset file in the file system. A system's effective launch against attacks will typically be much smaller than its full capacity to fight. To illustrate why this might be true, Let us consider 4 binary valued configuration of the hypothetical program such as: c, d, b, and a. Let us assume that all 16 possible phishing attacks of these options are valid. Also assume that for the system's test suite and with the testing goal of 100 percent line coverage the program consists of three interactions. Through single concrete configuration, all three interactions are satisfied, namely a ^ b ^ c ^ d. Thus, for this system, coverage goal and test suite, the effective antiphishing attack contains only one configuration, while the full phishing data has 16. Moreover, since at least one of the system's interactions involves three options, covering arrays of strength 2 or less would not be guaranteed to achieve maximal coverage, while covering arrays of strength 3 or higher attack models will contain efficient methods that add to overall coverage.

CLUSTERING

Given N item set to be clustered, the similarity matrix whose distance is N*N where the hierarchical clustering process undergoes here. The steps involved in the following are:

Step-1: Assign each item to each cluster. For example, if you have N items, then N clusters will be there because each cluster holds each item in it. Now, the distance between the clusters is same as that of the distance between the items.

Step-2: In order to less one cluster, merge the most similar (closest) cluster pairs into a single cluster.

Step-3: Compute the distances between each old cluster and new clusters.

Step-4: Repeat the above step 2 and step 3, until N size is equal to single cluster where all items are clustered.

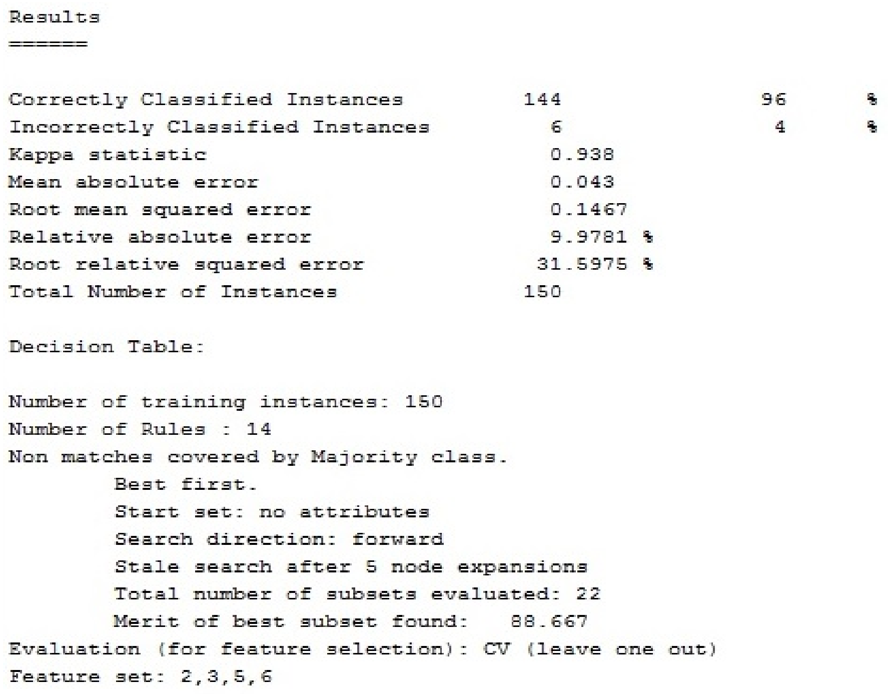
Step-3 is happening in different ways. From *complete* and a*verage-linkage* clustering, *single-linkage* clustering is distinguished. *Minimum* method is nothing but *single-linkage* clustering, which is defined as the distance between any member in one cluster to any member in the other cluster is same as that of the distance between one cluster to another cluster.

If the data contain similarities, one cluster to another cluster similarity is equal to any member in one cluster to any member in other cluster greatest similarity. In *maximum* method (i.e.) *complete_linkage* clustering, consider one cluster to another cluster distance is same as that of any member in one cluster to any member of in another cluster greatest distance.

On *moderate method* (i.e.) *average-linkage* clustering, consider one cluster to another cluster distance is same as that of average distance. This hierarchical clustering is called as *agglomerative.* Also, *advisive* hierarchical clustering subdivides a single cluster in small pieces and cluster object sequence, reverse execution is possible. Hence, this divisive method is rarely applied.

DECISION TREES - iTREE - CLASSIFIER

The clustering algorithms which are based on the minimum spanning tree (MST) doesn't assume the data points which are gathered across centers or the data points which is separated by geometric curve. Hence, in practice, it is widely used. MST partitioning and representative features selection is involved in this step. The relevant correlation measures are applied in correlation; hence the irrelevant features are removed. After removal the MST is then composed of two components which are connected in turn with the data set of irrelevant features and this process is called redundant feature elimination. The process will happen to extract the relevant features for target concept and also to eliminate irrelevant features. By choosing representative from distinct feature cluster, the latter removes the redundant features. Hence the final subset is produced through it. Loop algorithm iterates until it meets developer's supplied stop criteria (e.g., Expiration of time limit or No achievement of more coverage). This algorithm begins with an interactive iTree where tree contains one node, (i.e.) true. All configuration set which has been executed till now and their coverage information was recorded by iTree. In order to explore the next, the various heuristics pick up the leaf node which was used by BestLeafNode. Hence we have to find these BestLeafNode. We can't expect to explore full interaction tree, hence this heuristic is important. The output of the iTree classifier was shown in the Figures 3a, 3b, 3c.



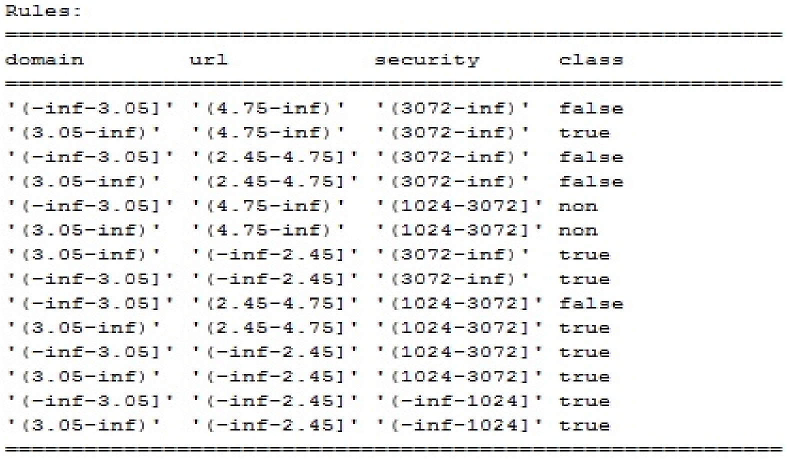
```
Results
======

Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
Kappa statistic                          0.938
Mean absolute error                      0.043
Root mean squared error                  0.1467
Relative absolute error                  9.9781 %
Root relative squared error             31.5975 %
Total Number of Instances              150

Decision Table:

Number of training instances: 150
Number of Rules : 14
Non matches covered by Majority class.
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 22
        Merit of best subset found:   88.667
Evaluation (for feature selection): CV (leave one out)
Feature set: 2,3,5,6
```

Figure-4. Classification accuracy of iTree classifier.



```
Rules:
===========================================================
domain         url            security       class
===========================================================
'(-inf-3.05]'  '(4.75-inf)'   '(3072-inf)'   false
'(3.05-inf)'   '(4.75-inf)'   '(3072-inf)'   true
'(-inf-3.05]'  '(2.45-4.75]'  '(3072-inf)'   false
'(3.05-inf)'   '(2.45-4.75]'  '(3072-inf)'   false
'(-inf-3.05]'  '(4.75-inf)'   '(1024-3072]'  non
'(3.05-inf)'   '(4.75-inf)'   '(1024-3072]'  non
'(3.05-inf)'   '(-inf-2.45]'  '(3072-inf)'   true
'(-inf-3.05]'  '(-inf-2.45]'  '(3072-inf)'   true
'(-inf-3.05]'  '(2.45-4.75]'  '(1024-3072]'  false
'(3.05-inf)'   '(2.45-4.75]'  '(1024-3072]'  true
'(-inf-3.05]'  '(-inf-2.45]'  '(1024-3072]'  true
'(3.05-inf)'   '(-inf-2.45]'  '(1024-3072]'  true
'(-inf-3.05]'  '(-inf-2.45]'  '(-inf-1024]'  true
'(3.05-inf)'   '(-inf-2.45]'  '(-inf-1024]'  true
===========================================================
```

Figure-5. Rules generated in iTree classifier.

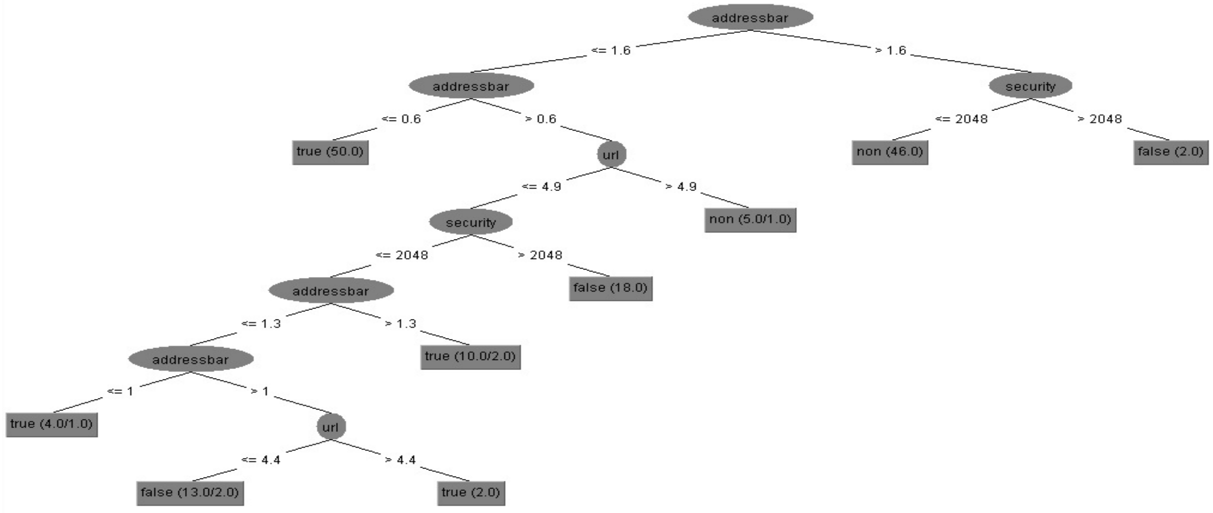


Figure-6. iTree Classifier's decision Tree.

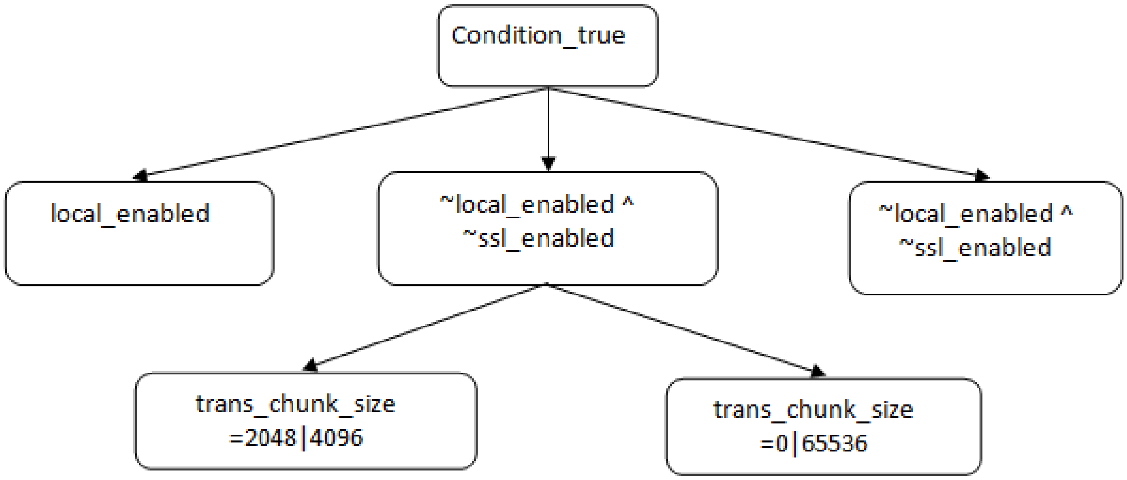
CONSTRUCT SPANNING TREE (MINIMUM)-iTree DISCOVERY

In order to catch the clustering algorithm Spanning Tree (Minimum) is used. But it doesn't assume that the data points are gathered across centers or the data points are separated by geometric curve. In this module, MST is constructed from the relevant features of the dataset by eliminating the irrelevant features. Among this relevant feature take a pair of features and apply feature correlation matrix in-between these features. This calculates the distance and thereby the information gain is attained using the entropy technique. After calculating the information gain of the pair of features, then choose a high information gain feature while the other features are removed from the relevant ones. As a result, we get the highly correlated features separately with target concept. Now construct the MST from these relevant features using the decision tree algorithm.

TREE PARTITION AND BUILDING

The tree partition or clustering module is used to eliminate the redundant features. Using graph-theoretic clustering method, the features are partitioned into clusters after the minimum spanning tree was constructed. This is also called tree partitioning. The tree will be partitioned below constraint. The constraint is taking a pair of features in the minimum spanning tree. Calculate the information gain of these features. If the information gains of these features are less than the target concept, then start partitioning. A cluster contains all features. Every cluster is an independent one. Each cluster is taken as a single feature. The next node in the tree is used to select the representative feature in each and every cluster. We select the representative feature from each cluster based on information gain. It means choosing the high information gain feature from each cluster. Finally, we get highly correlated feature subset from high dimensional data.

From this one can infer the correct design configurations.



The tree structure is constructed in the graph and the decision model is shown visually. The landings and the nodes with the splits are displayed. The configurations can thus be extracted. Control flow guards are stacked up each other because of this interaction arises in implementation terms, i.e., in addition to the lower strength interactions, additional constraints are also added by higher-strength interactions, which include option settings without discrimination, especially when it gets large.

EVALUATION RESULTS

The proposed clustering based decision tree is very accurate in finding the correct rules and configurations as it removes redundant and irrelevant features in the antiphishing training model. It also overcomes the dimensionality curse problem. Further, the capacity to handle high dimensional datasets is an added advantage. It does not buckle under the dimensionality curse. Each cluster found in the configuration dataset was taken as single feature the problem of dimensionality is reduced drastically. Given an input antiphishing data set, the ideal scenario would produce the rule set which reduces the faults for detecting the fake websites and improves the maintenance and this is what precisely this model does. The proposed iTree decision tree model provides the relevant system configuration which is easy to understand and has a visual tree like structure. The data elements are phishing attributes which form the clusters and hence the decision trees. The major problem of computational overhead is overcomed in this method. This proposed model of a new iTree based visual decision tree clustering algorithm provides the cluster's pruning and merging overlapping rate. The general experience of the resultant decision tree set and the phishing data sets show that the proposed method considerably decreases the time, computational cost, improves cluster's accuracy and space complexity reduction. The proposed algorithm measures resultant metrics in order to show advantages in software configuration clustering. Identifying the correct combination in a natural way and implementing the cluster's basic requirement are provided by resultant minimum spanning tree clustering algorithm. The ensuing decision tree shows the similarity in the form a visual decision tree to choose the right tools to fight phishing.

Figure-7 summarizes that the considered algorithm's classification accuracy (%) for the phishing problem set.

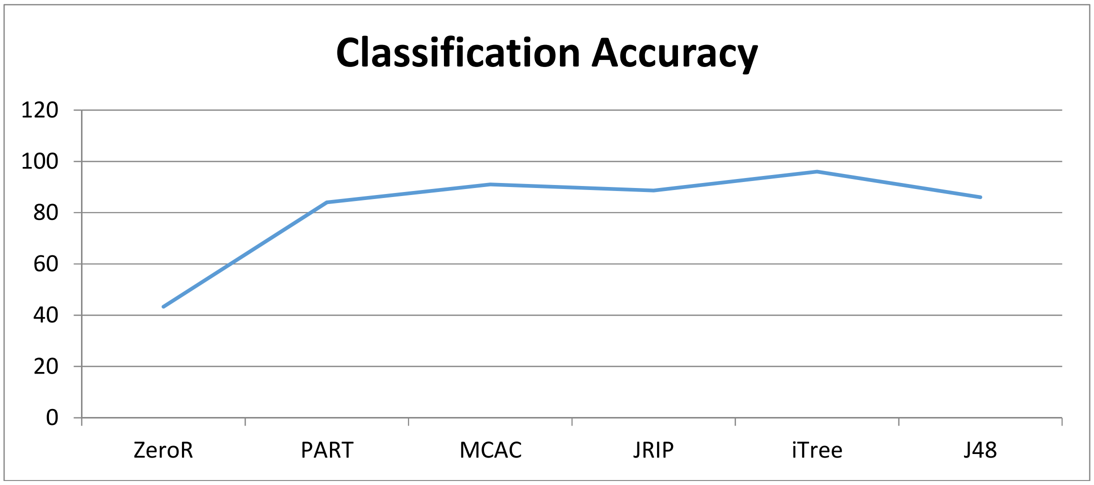


Figure-7. iTree algorithm outperformed well. In particular, iTree algorithm outperformed the J48, RIPPER, ZeroR, PART, and MCAC algorithms with 0.6%, 1.01%, 1.26%, 4.76%, 0.8%, respectively.

Overall, except zero, all algorithms' prediction accuracy is acceptable. Figure-5 displays all algorithms generates the number of rules from phishing data problem. Figure-8 shows that MCAR algorithm generates the number of rules. During the training phase, by evaluating each correlation between the class values and attribute values and also to learn rules, training case can be used once more in order to cause the large classifier.

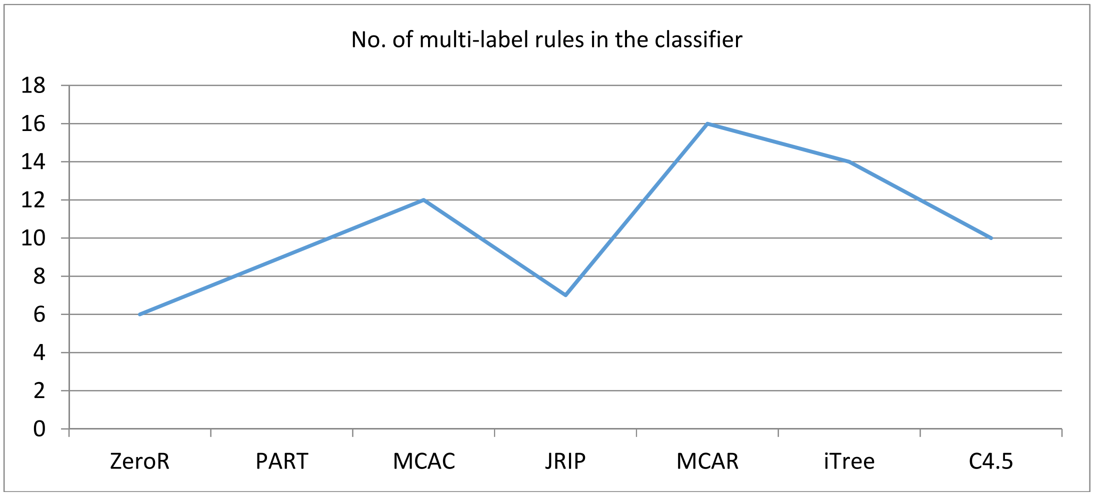


Figure-8. Displays the total number of multi-label rules based on the label of the class.

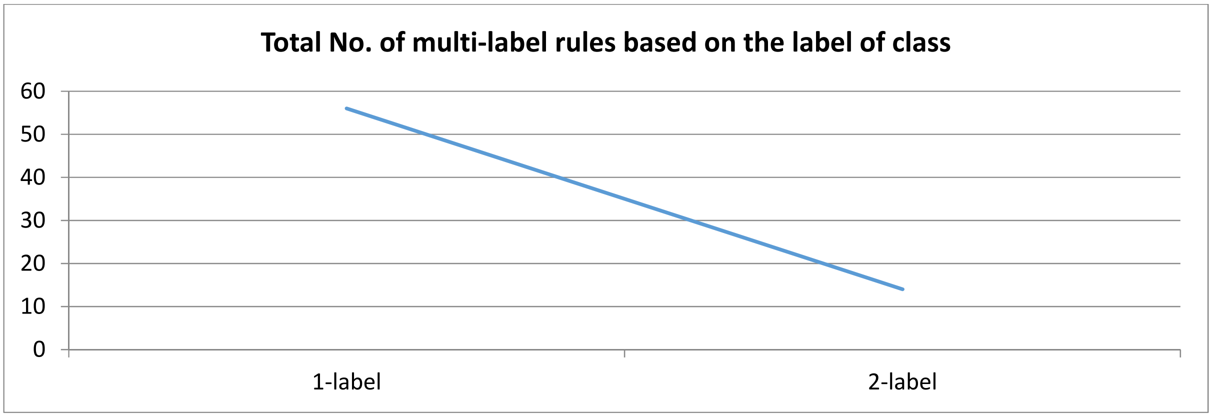


Figure-9. Shows that, iTree algorithm generates 14 multi-label rules that represents the two classes (i.e) "Phishy" or "Legitimate" class. Most of the classification algorithm (including current algorithm) classifies the test data as "Phishy" by linking these rules to the suspicious website.

CONCLUSION AND FUTURE WORKS

Thus in our hybrid model one of the classification an iTree model has proved that it is able to identify and fight against phishing with more accuracy using a specialized filter iTree. The iTree contains features extracted than existing models. Thus detecting the phishing websites is more crucial than other problems because of the nature of financial and other transactions done. An association based decision tree Classification is an accurate and intelligent approach that is also practical because it derives simple classifiers. The problem of website phishing has been solved in this paper where a data mining model is used to discover feature correlations and then generate effective rules in a simple manner. Unlike other existing methods this method diagnoses the new rules that are connected with one more class which gives the user, new type of arms to fight against phishing. To detect the phishy websites with more accuracy, these rules enhance the classification accuracy. Using frequency analysis, this model also identified significant features that relate to phishing websites. Thus the proposed authentication scheme works well against phishing techniques and protects the user data present in website. The computational overhead is also very less when compared with the existing techniques. In future works the project can be enhanced to be implemented for web services and in mobile environments where applications are densely concentrated, thus preventing phishing attacks robustly.

REFERENCES

[1] Millersmiles 2011. Millersmiles. <http://www.millersmiles.co.uk/>.

[2] D. Miyamoto, H. Hazeyama and Y. Kadobayashi. 2008. An evaluation of machine learning-based methods for detection of phishing sites. Australian Journal of Intelligent Information Processing Systems, 2: 54–63.

[3] G. Aaron, and R. Manning. 2012. APWG phishing reports. <http://www.antiphishing.org/resources/apwg-reports/>.

[4] David Kovarik. 2012. NUIT ecommunicator Fall Student Edition. Northwestern Journal of Technology and Intellectual Property. 2(1): 3-4.

[5] PhishTank. 2006. PhishTank. <http://www.phishtank.com/>.

[6] Allan Henry. 2012. Email Spoof Attack. Article of Life hacker.

[7] E. Mykletun, M. Narasimha and G. Tsudik. 2006. Authentication and integrity in outsourced databases. Trans. Storage. 2(2): 107-138.

[8] Phishing and Social Engineering. 2014. IT Secure Computing of Stanford University. <http://www.stanford.edu/>.

[9] Juels and B. S. Kaliski Jr. 2007. Pors: proofs of retrievability for large files. In CCS '07: Proceedings of the 14th ACM conference on Computer and communications security. New York, NY, USA: ACM. pp. 584-597.

[10] 2005. An Extendable Spam Filter System. At 2nd Conference on Email and Anti-Spam (CEAS), Stanford University, Palo Alto, California, USA.

[11] A.K. Jain, M.N. Murty, P. J. Flynn. 1999. Data Clustering. ACM Computing Surveys (CSUR). 31: 264-323.

[12] Min Wu, Robert Miller, Simson Garfinkel. Do Security Toolbars Actually Prevent Phishing Attacks? Citations: 132-1 self.

[13] Apache Software Foundation. Spamassassin homepage, 2006. <http://spamassassin.apache.org/. >

[14] Kevin Townsend. 2012. Whitelisting Vs Blacklisting. Security issues. <https://kevtownsend.wordpress.com/2011/08/24/whitelisting-vs-blacklisting/. >

[15] Muntajeeb Ali Baig. 2012. Effect of Social Networking Sites in Communication of Information in Distance Education. European Journal of Open, Distance and E-Learning.

[16] L. Breiman. 2001. Random forests. Mach. Learn. 45(1): 5-32.

[17] 2012. The EM Algorithm for Gaussian Mixtures. UCI Donald Bren School of Information and Computer Sciences. pp. 1-4.

[18] Alsaid and C. J. Mitchell. 2005. Installing fake root keys in a PC. On Euro PKI. pp. 227-239.