IJETIE
International Journal of Emerging Technology and Innovative Engineering
Volume I, Issue 4, April 2015
ISSN: 2394 - 6598
www.ijetie.org

Predicting Phishing Websites using Rule Based TECHNIQUES

O.Kalaiselvan, PG Scholar, P.S.R.Engineering College, Sivakasi,
S.EdwinRaja, Assistant Professor, P.S.R.Engineering College, Sivakasi.

***Abstract*— Recently, one of the great threats of spam related activities is that of phishing, particularly focusing on the detection of phishing website Phishing is considered a form of internet crime that is defined as the art of creating a website of an honest enterprise aiming to acquire confidential information such as usernames, passwords, etc. There are few characteristics that distinguish phishing sites from original ones such as Length of the URL is long, IP address in URL, join prefix and suffix to domain and request URL, etc. Even though there are numerous methods reported to avoid Phishing each method has its own limitations. Our aim is to build a group of features that have been shown to be sound and effective in predicting phishing websites and to extract those features according to new well-defined rules. We propose an anti-phishing method to safeguard Internet users from the phishing attacks.**

***Index Terms*— Phishing Attacks, Rule based Classification, Toolbar, web spoofing, User Protection.**

I. INTRODUCTION

Phishing as the act of stealing personal information of Internet users for misuse is an old but still threatening problem. It is a form of identity theft that occurs when a malicious Web site impersonates a legitimate one in order to acquire confidential information such as account details, user passwords, etc. Though there are several anti-phishing software's and techniques for detecting potential phishing attempts in emails and detecting phishing information on web pages, Phishers come up with new and reduction techniques to circumvent the available software and techniques. The phishing attacker's mislead users by employing different social engineering tactics such as threatening to suspend user accounts if they do not complete the account update process, furnish other information to validate their accounts or some reasons to get the users to visit their spoofed web sites. Why is it important to tackle the concern of phishing? According to the Anti-Phishing Working Group[1], there were 18,480 unique phishing attacks and 9666 unique phishing sites reported in March 2006. Phishing attacks affect millions of internet users and are a huge cost burden for businesses and victims of phishing. Phishing has become a serious threat to users and businesses alike. Over the past few years, plenty of attention has been paid to the issue of security and confidentiality. Existing literature handle with the problem of phishing is scarce.

As the amount of Internet users and online transactions multiples, the possibility of misuse is also growing. Phishing is hence an important cyber security issue. Projecting this to a whole year more than three million URLs are documented each year. Nowadays, phishers use sophisticated software toolkits to launch a large number of Phishing websites on different URLs to counteract common security methods like blacklist that are most commonly used as phishing protection. Users could sight most of the phishing attacks by themselves by closely examining URLs and other indicators with the right amount of security awareness and focus, but as security is never the users' main goal [8] they fail to detect most of the attacks. Another issue is that phishers usually try to closely impersonate a trusted party the user knows by imitating brands, website design, logos or as a special case the URLs. This being an issue for users to fall for phishing, it should be used as an input to online security research to generate new means for phishing analysis. In our paper, we study the common procedure of phishing attacks and review possible anti-phishing approaches. In this work we want to focus on URL similarity.

Phishing attack classically starts by sending an email that appears to come from an enterprise to victims asking them to update or confirm their personal information by visiting a link within the email. Although, phishers are now using several techniques in creating phishing sites, they all use a set of mutual features to create phishing websites since without those features they lose the advantage of deception. This helps us to differentiate between honest and phishy websites based on the features extracted from the visited website. Since attackers usually cannot use the exact same URL they are aiming, they use different deceptions to build domain names and paths that look similar to the original. For example they use small spelling mistakes that might be unnoticed by the user. In case the spelling of a phishing URL is close to a real domain name or brand name automatic detection of phishing attacks becomes available. We present such a detection approach using URL terms together with search engine spelling suggestions.

. Overall, two approaches are used in identifying fraudulent web sites. The first method is based on a blacklist is, in which the requested URL is compared with those already created list.

The drawback of this approach is that the blacklist usually cannot cover all phishing web pages since, within few seconds, different fraudulent web page is hosted. The second method is heuristic-based methods, where several features are collected from the website to categorize it as either phishy or legitimate. When compared to the blacklist approach, a heuristic-based method can recognize freshly created phishing web pages.
The accuracy of the heuristic-based methods depends on picking a set of specific features that could help in differentiating the type of web page. The way in which the extracted features also plays a major role in classifying websites accurately.

II. PROBLEM STATEMENT

Phishing web pages are fake websites generated by dishonest people to impersonate original web page. Users may not be able to access their emails or sometimes lose money because of phishing. Predicting and blocking this attack is a critical step toward protecting online transactions. The efficiency of predicting the type of the website necessarily depends on the extracted features goodness. Since most of the users feel safe against phishing attacks if they utilize an anti-phishing tool, this deliver a great responsibility on the anti-phishing tools to be accurate in predicting phishing.

In that situation, we consider that developing rules of thumb to extract specific features from websites then utilizing them to predict the type of web page is the key to success in this case.

III. LITERATURE REVIEW

Phishing website is a huge effect on the financial and online transactions, detecting and preventing this attack is an important step towards protecting against website phishing attacks, there are many approaches to detect these attacks. In this section, we review about existing anti phishing solutions and list of the related study.

One approach is a **client-side defense against web-based identity theft [2]**. It proposes a framework for client-side defense: a browser plug-in called Spoof Guard that examines web pages and warns the user when requests for data may be part of a spoof attack, it calculates a spoof index (a measure of the likelihood that a specific page is part of a spoof attack), and alert the user if the index exceeds a level selected by the user. Spoof Guard uses both combination of outgoing post data examination to compute a spoof index and page evaluation. When a user enters a username and password on a spoof website that contains some combination of suspicious misleading domain name, URL, images from an honest website, and password and a username that have previously been used at an honest website, Spoof Guard will block the post and warn the user with a popup that foils the attack.

The paper describes common properties of ten spoof websites recently found, they are Suspicious URLs, Logos, User input, Copies, Short lived, HTTPS and Sloppiness or lack of familiarity with English. The browser plug-in applies analysis to all downloaded pages and combines the results using a scoring method. The total spoof index of a web page determines whether the plug-in warns the user and determines the severity and type of alert. Since popup messages are intrusive and disturbing, it attempts to alert the user through a passive toolbar indicator in most of the case.
In order to apply image and URL check, the Spoof Guard plug-in is provided with a fixed database of images and their related domains. When the web browser downloads a login page all images on the page are compared to images in the Spoof Guard database. The spoof-count for the page is increased if a match is found but the page's domain is not a original domain for the image. The web browser previous history record and additional history stored by Spoof Guard are used to evaluate the citing page. When a user enters in form data, the Spoof Guard block's and checks the HTML post data, allowing the absolute post to progress only if the spoof index is below the user specific threshold for posts.

Second approach is **Anomaly Based Web Phishing Page Detection [3]**. This checks the anomalies in web pages, in particular, the variation between a web site's identity and HTTP transactions and its structural feature. A structured page is composed of W3C DOM objects. Among them, it scrolls five categories, based on their importance to the web identity. They are Server Form Handler (SFH), Request URL (RURL), Keyword/Description (KD), Main Body (MB), and URL of Anchor (AURL). The above categories are the important sources which the identity and features are derived from and lists the characteristics of phishing like Abnormal DNS record, Abnormal Anchors,, Abnormal URL, Abnormal certificate in SSL, Abnormal Server Form Handler, Abnormal cookie, Abnormal Request URL.

It extracts the relevant web objects from a webpage and converts them into a feature vector based on the of phishing analysis. The page classifier takes the feature vector as input and determines whether the page is bogus or not. The proposed phishing detector consists of two components Page Classifier and Identity Extractor.

Identity Extractor individually identifies the web pages ownership; the identity is a unique string appearing in its domain name and/or an abbreviation of the organization's full name.

Page classifier specifies to these objects/properties as structural features. One source of the structural features is those identity relevant W3C DOM objects in a web page, Example, URI (Uniform Resource identifier) domain of an anchor, and another source of the structural features is HTTP transactions. The page classifier here employs SVM (Support Vector Machine), a very well known algorithm for classification. It then results as label 1 which indicating a phishing web page or a label -1 which indicating an authentic one.

To facilitate SVM (Support Vector Machine) based classification, they compute those features into vectors. The result from the execution of the page identity extractor is a character strings from cited identity words. The feature vector computation of webpage are certificate in SSL, DNS record ,URL of anchor , request URL , server form handler , URL address, and domain in cookie. Given an identity and a set of features, the main task of determining the genuineness of a page is executed by SVM (Support Vector Machine), which is a very well known classifier and has been widely employed in pattern recognition.

In this approach [4], one approach employed here is based on experimentally contrasting associative classification algorithms. The authors have gathered different features from various websites. Those features ranged among three fuzzy set values "Legitimate, Genuine and Doubtful". To evaluate the selected features, the authors conducted experiments using the following data mining techniques, MCAR, CBA, C4.5, PRISM, PART and JRip. The results showed an important relation between "Domain Identity" and "URL" features. There was insignificant impact of the "Page Style" on "Social Human Factor criteria"

IV. ALGORITHMS

The algorithms used here to analysis are listed below:

ID3 Decision Tree

ID3 is a simple decision tree learning algorithm developed by Ross Quinlan [4]. The idea of this ID3 algorithm is to construct the decision tree by employing a top-down, greedy search through the given data sets to test each attribute at every tree node, and in order to select the attribute which is most useful for classifying a given data sets. A statistical property called information gain is defined to measure the worth of the attribute.

C4.5 DECISION TREE

C4.5 algorithm [5] is a successor of ID3 algorithm that uses gain ratio as splitting criterion to separate the data set. The algorithm implements a kind of normalization to information gain using a split information value.

A decision tree is a predictive machine-learning model that decides the target value (dependent variable) of a new sample based on various attribute values of the available data. The internal nodes of a decision tree denote the different attributes; the branches between the nodes tell us the possible values that these attributes can have in the observed samples, while the terminal nodes tell us the final value (classification) of the dependent variable. The attribute that is to be predicted is known as the dependent variable, since its value depends upon, or is decided by, the values of all the other attributes. The other attributes, which help in predicting the value of the dependent variable, are known as the independent variables in the dataset.

CART DECISION TREE

CART [11] stands for Classification and Regression Trees introduced by Brieman. It is also based on Hunt's algorithm. CART handles both categorical and continuous attributes to build a decision tree. It handles missing values.

CART uses Gini Index as an attribute selection measure to build a decision tree. Unlike ID3 and C4.5 algorithms, CART produces binary splits. Gini Index measure does not use probabilistic assumptions like ID3, C4.5. CART uses cost complexity pruning to remove the unreliable branches from the decision tree to improve the accuracy.

ZeroR Classification algorithm

ZeroR [12] classifier predicts the majority of class in training data. It predicts the mean for numeric value & mode for nominal class.

V. ANALYSIS

Details of data Set:

We used dataset for evaluation with classifier on WEKA. A set of phishing web page was collected from the Phistank [6] website, which is a free community web page where users can submit, track, share phishing data and verify. We have collected 1000 phishing URL's. The dataset is described by the types of attributes, the number of instances stored within the dataset, the data type being used; also the table demonstrates that all the selected data sets are used for the classification task. The datasets were chosen because they have different characteristics and have addressed in different areas. Dataset is in CSV(Comma-Separated Values)format. Dataset have 17 numbers of attributes. The below table 1 shows the details of the data set.

Name of Data Set	Type of file	Number of Attributes	Number of instances	Attribute characteristics	Dataset Characteristics
Phishing Data Set	CSV(comma separated value)	17	505	Nominal	Nominal

Table 1: Data Set

Comparative Analysis:

We compare different rule classification algorithms, each of which utilizes a different methodology in producing knowledge. Using the WEKA knowledge flow environment the model is created. The below Figure 1. Shows the comparative model

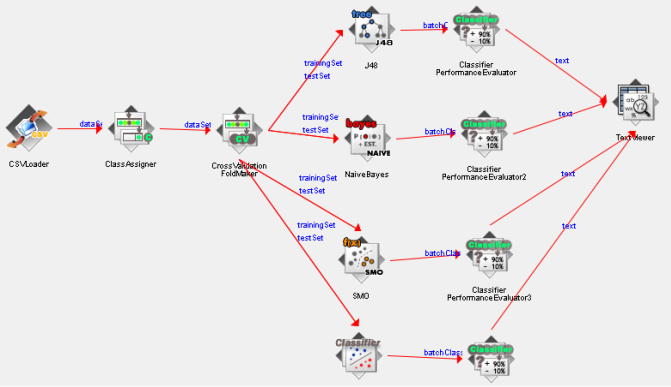

Figure 1: Block Diagram

In the classify panel we choose ZeroR classifier and start the analysis using 10 fold cross validation. The result for ZeroR classifier is shown below in the figure 2.

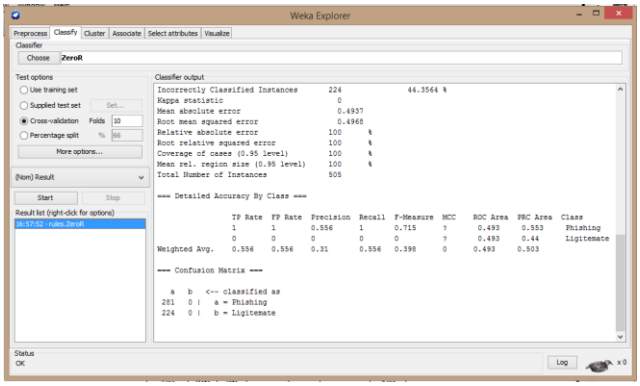

Figure 2: ZeroR

In the classify panel we choose C4.5 classifier and start the analysis using 10 fold cross validation. The result for C4.5 classifier is shown below in the figure 3.

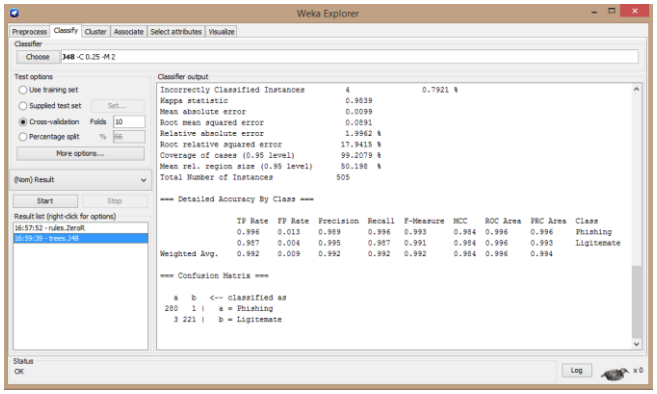

Figure 3: C4.5

In the classify panel we choose NaiveBayes classifier and start the analysis using 10 fold cross validation. The result for NaiveBayes classifier is shown below in the figure 4

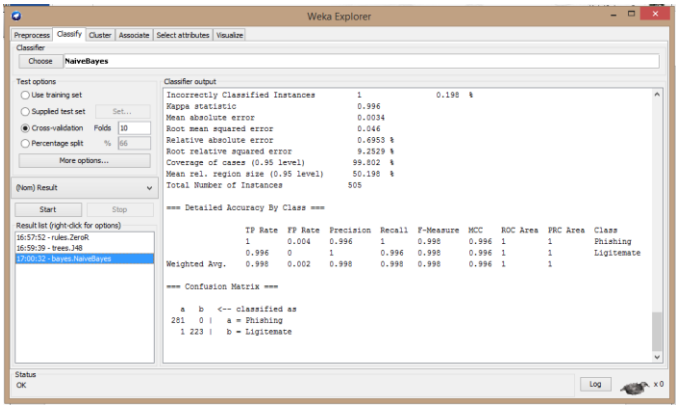

Figure 4: Naive Bayes

In the classify panel we choose SVM classifier and start the analysis using 10 fold cross validation. The result for SVM classifier is shown below in the figure 5

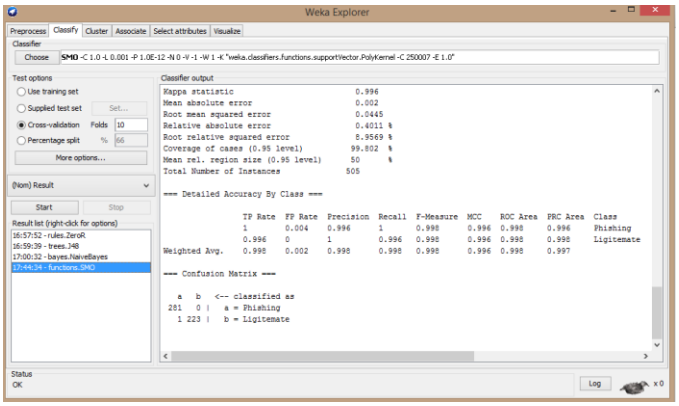

Figure 5: SVM

.

I tried to evaluate the performance of various classifiers on test mode 10 fold cross validation with data sets at WEKA 3-7-7, The results after evaluation is described here in the below mentioned table 3

Table 2: Evaluation of Classifiers

Classifier	Classifier Model	Test Mode	Correlation Coefficient	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Root Relative Squared Error
C4.5	Full Training Set	10 Fold Cross - Validation	0.9839	0.0099	0.0891	1.9962%	17.9415 %
Naïve	Full Training Set	10 Fold Cross - Validation	0.996	0.0034	0.046	0.6953%	9.2529%
SVM	Full Training Set	10 Fold Cross - Validation	0.996	0.002	0.0445	0.4011%	8.9569%
Zero R	Full Training Set	10 Fold Cross - Validation	0	0.4937	0.4968	100%	100%

To classify them correctly from the training data set the error rates and accuracy using classifiers are evaluated. Below

table 4 shows the final statistic of different classifier.

Class-ifier	TP Rate	FP Rate	Precision	Recall	F - Measure	ROC Area	Class
C4.5	0.996	0.013	0.989	0.996	0.993	0.996	Phishing
	0.987	0.004	0.995	0.987	0.991	0.996	Ligitemate
Naïve	1	0.004	0.996	1	0.998	1	Phishing
	0.996	0	1	0.996	0.998	1	Ligitemate
SVM	1	0.004	0.996	1	0.998	0.998	Phishing
	0.996	0	1	0.996	0.998	0.998	Ligitemate
ZeroR	1	1	0.556	1	0.715	0.493	Phishing
	0	0	0	0	0	0.493	Ligitemate

Table 3: Final Statistic of different classifier

The error rates of various classifiers are compared and this is shown in the below table 5.

Classifier	TP Rate	FP Rate	Precision	Recall	F - Measure	ROC Area
C4.5	0.992	0.009	0.992	0.992	0.992	0.996
Naïve	0.998	0.002	0.998	0.998	0.998	1
SVM	0.998	0.002	0.998	0.998	0.998	0.998
ZeroR	0.556	0.556	0.31	0.556	0.398	0.493

Table 4: Comparison of weighted average for different classifier

.

From the below graph (figure 6), it is observed that Naïve Bayes algorithm performs better than other algorithms. Therefore the Naïve Bayes classification algorithm performs well because it contains highest accuracy when compared to C4.5, SVM and ZeroR.

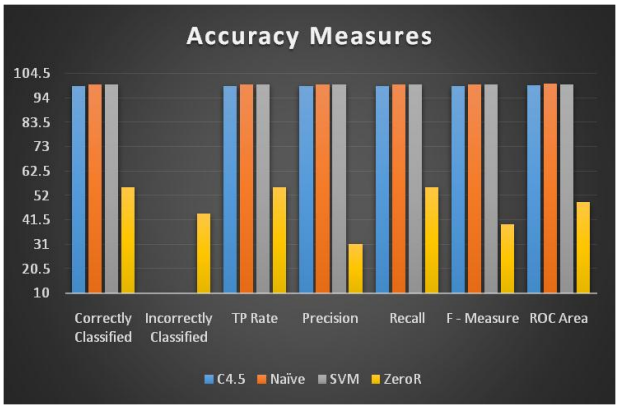

Figure 6: Accuracy measures for different classifier

.

We also compare the confusion matrix for different classifiers. The comparison table is shown below in the table

Classifier	a	b	Parametric Variable	Recognition
C4.5	280	1	a	99.21%
	3	221	b	0.79%
Naïve	281	0	a	99.80%
	1	223	b	0.20%
SVM	281	0	a	99.80%
	1	223	b	0.20%
ZeroR	281	0	a	55.64%
	224	0	b	44.36%

Table 5: Confusion matrix for different classifier

From the below graph (figure 6.15), it is observed that C4.5 and Zero R algorithms attains highest error rate. Therefore, the Naïve Bayes and SVM classification algorithm performs wellbecause itcontains least error rate when compared to other algorithms.

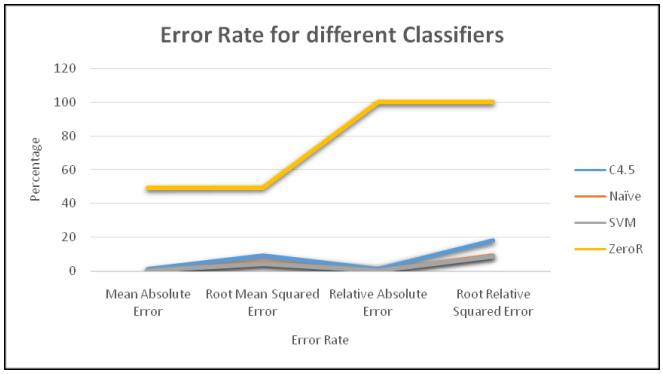

Figure 7: Error rate for different classifier

VI. CONCLUSION

Here, the proposed approach gives proper result for detecting fake website. Classification is one of the most popular techniques in data mining. In this paper we compared several algorithms. Our measure of interest includes the analysis of classifiers on Phishing dataset, the results are described in value of correctly classified instances & incorrectly classified instances (for dataset with nominal class value), mean absolute error, root mean squared error, relative absolute error, root relative squared error after applying the cross-validation. Through our experiment we conclude that Bayesian algorithms have good classification accuracy over other compared algorithms. In the near future, we will use the rules produced by different algorithms to build a tool that is integrated with a web browser to detect phishing websites on real time and warn the user of any possible attack.

REFERENCES

[1]Anti-Phishing Working Group. Phishing Activity Trends Report, http://docs.apwg.org/reports/apwg_trends_report_q1_2014.pdf, 2014

[2]N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. C. Mitchell, "Client–side defense against web–based identity theft", In Proceedings of 11th Annual Network and Distributed System Security Symposium, 2004.

[3]Y. Pan and X. Ding, "Anomaly Based Web Phishing Page Detection", Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC'06), Computer Society, 2006

[4] Rami M. Mohammad, Fadi Thabtah, Lee McClusky, "Intelligent Rule based Phishing Websites Classification," in IET Information Security, Volume 8,Issue 3, pp. 153–160.2014

[5] J.R.Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers, Inc, 1992.

[6] "PhishTank Home," http://www.phishtank.com/. [Online]. Available: http://www.phishtank.com/

[7] Anshul Goyal, Rajni Mehta, "Performance Comparison of NaïveBayes and J48 Classification Algorithms"

[8]Dhamija, R., and Tygar, J. (2005) The battle against phishing: Dynamic security skins. In Proc. ACM Symposium on Usable Security and Privacy (SOUPS 2005), (pp. 77–88)

[9] Abdullah H. Wahbeh, Mohammed Al-Kabi, "Comparative Assessment of the Performance of three WEKA text classifiers applied to Arabic Text"

[10]SJ, Fan P, Khan MK, Run RS, Lai JL, Chen RJ, et al. An efficient phishing webpage detector. Expert Systems with Applications: An International Journal. 2011; 38 (10): p. 12018-12027

[11]J. R. Quinlan, "Introduction of decision tree", Journal of Machine learning", : pp. 81-106, 1986

[12] Sunita B.Aher, Lobo L.M.R.J, "Comparative study of classification algorithm" in International Journal of Information Technology and Knowledge Management, volume 5, No.2, pp. 239-243.