Asian Research Consortium

Asian Journal of Research in Social Sciences and Humanities
Vol. 6, No. 11, November 2016, pp. 350-366.

Asian Journal of Research in Social Sciences and Humanities

ISSN 2249-7315
A Journal Indexed in Indian Citation Index

www.aijsh.com

DOI NUMBER: 10.5958/2249-7315.2016.01197.7

Category:Science and Technology

Consistency Testing of Phishing Websites based on Random Model Approach using Classification Algorithms

Emilin Shyni C*; Anesh D Sundar A**; Swamynathan S***

*KCG College of Technology,
Chennai, India.
**Karunya University,
Coimbatore, India.
***Anna University,
Chennai, India.

Abstract

Phishing is the act of attempting to trick people into parting with their sensitive information such as usernames, passwords, credit card and bank details, by way of email spoofing, instant messaging, or by using fake web sites whose look and feel make them seem legitimate. However, upon detailed examination, it is discovered that dealing with phishing websites is quite a challenging task. To overcome these problems, a method has been proposed and developed by the authors of this paper, for validating websites based on the Behavioral Random Model approach. This approach is defined by eight sets of features, which are in turn based on three types of Heuristics with a random number of inputs and corresponding responses. Subsequently, the most frequently occurring feature-combinations between phishing and legitimate websites were identified, in order to infer the integrity of each site. A tool named PhishDetector was implemented for automating the testing process. The results show the approach to have 100% true negatives and 0.03% false positives in detecting phishing and legitimate websites. The results were then classified on the basis of the classifier algorithms.

Keywords: Phishing, Heuristics, Legitimate website, PhishDetector, Classifier, Feature-Combinations.

1. Introduction

A phishing attack is a criminal activity which mimics a certain legitimate webpage using a fake one, with the intention of luring end-users to visit the fake website, and thereafter attempting to steal their personal information which includes usernames, passwords and details of their credit cards. There are several means of avoiding phishing, but research in this field is at a very nascent stage. In most cases, phishing involves attacking well-known web pages, and those that are in need of high-security. Therefore, the discovery and effective and efficient prevention of phishing websites, and identifying the targets of a phishing webpage are critical challenges. There are a variety of methods that can be used to identify a webpage as a phishing site, including whitelists (lists of known safe sites), blacklists (lists of known fraudulent sites) [19,20], Heuristics and community ratings. A number of approaches have been introduced in the last few years to fight phishing attacks. These include detecting suspicious websites with Heuristics, educating and training users, compiling whitelists and blacklists[1-2,16], filtering emails [3], and customizing visual cues to distinguish legitimate websites from fake websites. Most of the browsers have built-in phishing attack detection capabilities based on white and blacklisted websites [4].

In this paper, user-behavior was studied as an aid in the detection of phishing websites. The basis of this approach is the use of form submissions with a random number of inputs and several iterations of each of the pages. Eight different features were developed based on three types of Heuristics. The most frequently occurring different feature combinations were identified for the suspected phishing websites in comparison to the legitimate websites. Manual checking is required to decide, if no known heuristic combination is satisfied. Thus, a tool named PhishDetector was developed in order to automate the process of testing suspected phishing websites.

A supervised classification technique that is known as a major data-mining stream is used in this work, to verify and grade the severity of the phishing attacks. The modeling prediction task uses machine-learning algorithms, which infer the function from the supervised data on training. Some of the supervised learning algorithms including Decision Tree induction (DT), Multilayer Perceptron (MLP), Naïve Bayes (NB), Bayesian Network (BN) and Radial Basis Function (RBF), are used to produce an inferred function called the classifier, by analyzing the training data. For any valid unseen input object, the accurate output value is predicted using this classifier. Though various experiments have been conducted to measure the performance of these approaches, a promising approach for the resolution of this problem is through the Behavioral Response. This approach does not include the use of updated black or whitelists and is independent of the language and textual contents of the websites. Evaluation of this approach was carried out using both phishing and legitimate websites.

This paper is ordered as follows: - Section 2 illustrates literature survey on various methods of phishing-detection based on different approaches including the Heuristics approach, Section 3 describes the proposed architecture with eight features based on the forms. Section 4 discusses the implementation. Finally, section 5 arrives at certain conclusions and discusses the proposed work for the future.

2. Literature Survey

In recent years, many researchers have turned their attention to behavioral and trustworthiness of testing phishing websites. Y. Pan [5] proposed an anti-phishing scheme mainly built upon the detection of the identity relevant anomalies and securitizing how those anomalies were rated through DOM objects and HTTP transactions. The work consisted of an identity extractor and a page classifier. The identity of a website is defined as a set of words and is indicated with a number of objects or properties of the web page. If the features extracted from the phishing site are not matched with the original identity, then the website is considered suspect. The SVM classifier takes the vector as an input and phishing label as the output. The identity extractor has a lower success rate when processing the legitimate pages. This process neither requires online transactions nor requires users to change their navigation behavior. If the search engine returns an URL with the same domain, the web page under scrutiny is genuine with a crushingly high probability.

Xiang et al. [6, 15] pertain to information extraction and retrieval techniques to detect phishing pages. The DOM of a downloaded page is inspected to chart out its identity in the course of different attributes and the actual domain is classified based on identity. This hybrid based approach consists of an identity based detection component and a keyword detection retrieval component. The suspected page of the domain's result is then compared with the previous result. If the result set is a mismatch, the downloaded page is suspected to be a phishing page. The initial step is to locate entity names in DOM text nodes, which are most likely to represent the site's brand name, and then to find domains for that name by searching and comparing the matching domains with the page's domain itself, to determine identity inconsistency. The whitelists are collected from Google safe browsing, miller smiles, and white domain service.

Chou et al. [7, 17] presented the SpoofGuard tool that identifies phishing web pages based on Heuristics and computes a 'spoof value' based on matched heuristics. The features of stateless evaluation, stateful evaluation and input data include heuristics matching. If the value exceeds a threshold, a page is suspected for phishing. They propose a framework for client side protection. Stateless method determines whether a downloaded page is doubtful, whereas the stateful method evaluates a downloaded page in light of previous user activity. The Stateless method consists of URL check, Image check, Link check and Password check and the Stateful method consists of a domain check and referring page. The last method consists of outgoing password check, interaction with image check and a check with all posted data. This process takes advantage of unauthenticated emails and weak website authentication.

Wenyin et al. [8] presented a Semantic Link Network (SLN) for a set of nodes, and a link that connects two nodes (that is Nodes are considered as pages), calculating the implicit relationship between pages by a reasoning process. By using predefined rules and strategy, the phishing target of a phishing webpage can be determined. Calculating the association relations based on the three relation types such as link relation, text relation, and search relation, the Link relation is measured based on hyperlinks. The search relation from one page to the next page results in the content of the first page. However, if the text on a current page is very similar to that of the original webpage and if the domain names are different, the current page is suspected to be a phishing attempt. But this work does not take into account, the visual similarity relation, layout similarity relation and domain similarity relation. Ye cao et al [9] says that the usage of blacklists and whitelists is one of the best methods to prevent phishing. A Whitelist contains the URLs of the legitimate sites whereas a blacklist contains the URLs of the phishing sites. Frequent updates are a requisite for blacklists, failing which new phishing sites could slip in from time to time. Similarly, a whitelist also needs to be updated. Unfortunately such a list cannot contain all legitimate sites.

Nelson et al. [10] explains the method of exploiting a spam classifier to render it useless using a very specific attack framework, by using indiscriminate, focused attacking and an optimal attacking function. All of them assume that the training model used for the spam filter is based on the Naïve Bayes classifier. This paper speaks of the RONI defense which filters out dictionary attack messages with complete success. The dynamic threshold defense mitigates the effect of the dictionary attacks. The main idea is that the attack node first sends traffic that causes the Autograph to mark it suspicious, and then sends traffic similar to legitimate traffic, resulting in rules that cause denial of service.

According to Liu et al. [11], the phishing web pages are detected using a legitimate URL. Their paper describes the process of extracting the webpage's features and measures the similarity of the true pages according to three metrics: block-level (detail), layout (global), and style (overall). Earth Mover's distance method (EMD) is used to calculate the visual similarity of web pages. This method is image-based rather than HTML based, by which phishing webpage obfuscation scams are cracked. If the visual similarity is more than the resultant threshold, the method gives phishing information to the client. The suspicious URLs are generated based on heuristic rules. The drawback is that the Heuristics might not be able to generate all possible phishing URLs. Zhang et al. [12] developed the CANTINA tool which influences the TF-IDF (Term Frequency and Inverse Document Frequency) to categorize most weighted words and create signatures from the top five most important words. These signatures are identified using a search engine. CANTINA is compared with spoof-guard and Netcraft, and the results show that CANTINA provides better performance. They used the TF-IDF algorithm, in which TF counts the number of times the term is available in a document, and IDF measures the general importance of the term. True Positive and False Positive metrics are used to evaluate the result. The resultant domain names are compared with the current domain. A mismatch proves that the page is illegitimate.

Xin (Robert) Luo et al [13] proposed a model called Heuristic-Systematic Model (HSM), for information processing. This model integrates two information processing modes: Heuristic Processing and Systematic Processing. In this background of phishing attacks, fake posts are used with the purpose to deceive message receivers. Six heuristics have been analyzed with 105 people listed in a school's staff and faculty distribution lists. This paper supplies the theoretical support for future research in both qualitative and quantitative data and will be improved upon to inspect more meticulously and estimate research models and hypotheses.

Hossain Shahriar and Mohammad Zulkernine[14] proposed a behavior model approach which is described using the Finite State Machine (FSM) that collects the posted forms with random inputs and the resultant replies. Their tool name is PhishTester, which evaluates both phishing and legitimate websites. This approach has the ability to detect a few XSS-based phishing attacks as well. This approach does not handle random input submission to forms that contain captchas.

In compare to all these mechanisms, our approach is to confirm phishing attacks by studying the behavior of users while submitting a form, identifying the eight features and analyzing the data set with different classifiers.

3 System Architecture

This trustworthiness testing approach checks whether the behavior of the websites matches with the knowledge of phishing or legitimate website behavior to decide on its genuineness. It does not include the use of updated black or white lists, and is independent of the language and textual contents of the websites. It is described by capturing a form's submission with a random number of inputs and the corresponding responses. It provides users the flexibility to perceive phishing websites that may pilfer information by a tentative amount of pages having forms and utilizes different kinds of form generation methods. A supervised classification technique that is known as a major data-mining stream is used in this work, to verify and grade the severity of the phishing attacks.

The behavioural approach primarily involves several functional stages, namely, DOM parser, Random input generator, form poster, outbound analyser, content analyser, Timer and repository as shown in Fig1. The parser used here, is the Document Object Model (DOM) parser. The DOM interface is used to access and manipulate structured data. Several different object types are defined in the DOM specification. Each of these object types pertains to specific methods and properties.

The DOM parser initially parses all the content from the website which retrieves all the hyperlinks of the pages and the number of pages available in that website. It retrieves the list of pages, which contains the form fields and redirected links from the website, and stores it in the database. Random inputs are applied to the form fields, and the form is posted to find the maximum error that has occurred. The links which travelled from one page to the next page should be clearly noted, to find whether it is redirected to the same website or to a different one. This is due to the fact that phishing hyperlinks will be redirected to the legitimate website through the hyperlinks, to pilfer information from the legitimate website. Multiple posts of failures indicate the form's submission. The timer shows the total amount of time for parsing the whole website page. The content analyzer indicates the content of the form fields and the parsed links as shown in Figure 1. The legitimate website generates an error report for the random inputs, thereby checking if the error messages actually originate from the website or not.

Entering the form fields of the web pages assesses the behavioral response of the user. The number of times the forms are posted, the error messages given by the website, if any, and the feedback given by the website, are all considered for the response. For example, a phishing URL http://flaouz.com/gmail/gmail.html is taken from the phishtank, and given as the input to this PhishDetector tool. The flaouz.com website consists of four pages and these four pages are retrieved and are again tested to check whether they contain any forms or not. The first two pages contain no forms and the third and fourth page consist of forms. Those two pages are http://thestudy.my/dpo/ security.php and http://thestudy.my/dpo/tip.htm.These two pages are again tested and checked to find the form fields are available in them. Then, random inputs should be fed and checked with the number of times the forms are posted for per input. The legitimate website will give the error report for the random inputs, and hence, it is also a check to see if the error messages are generated from the website.

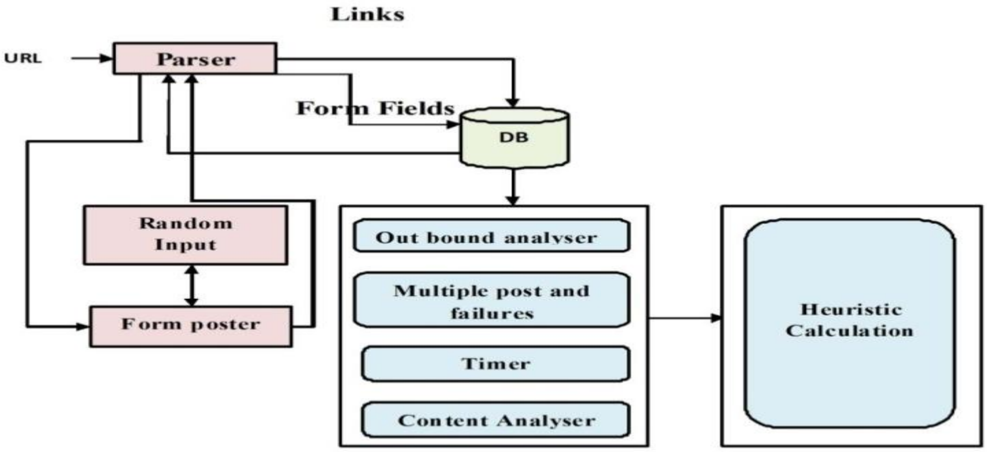


Figure 1. System Architecture of Phish Detector

Based on the analysis of the above results, additional testing has been carried out with the classification algorithms. To facilitate checking, several heuristics are checked based on the model to identify whether the website is a phishing or a legitimate one. Three different heuristics were then developed and it was found that each of the heuristics consists of different features.

3.1 Heuristics

The patterns in the URL or page are identified by the heuristics corresponding to suspicious websites. This process achieves better performance than blacklisting. One accepted technique aims to incorporate anti-phishing protection hooked on the web browser of the user, using two kinds of classification methods: blacklists and heuristic tests. The blacklists are simple in design and the implementation in the browser or in the application is easy, but the major problem is its incompleteness. The reason is that the attackers are very shrewd and may use many complicated techniques to escape blacklists. At some point, the incompleteness difficulty cannot be removed simply, since malicious URLs cannot be known before a certain quantity of frequency in the natural phenomena. Due to the intrinsic difficulty in the blacklist, this approach is implemented with Heuristics but without considering the blacklists. In order to test the effectiveness of the Heuristics, the system is designed with its own Heuristics-based anti-phishing tool, called the PhishDetector.

Based on the characteristics of the phishing URLs and web pages, three heuristics combination tests have been defined and implemented. Then the effectiveness of the Heuristics is experimented upon, thereby helping determine which of the Heuristics are important to differentiate the legitimate sites from the phishing ones. There are three types of Heuristics identified, namely, Link Analysis, Decision-based and Content-based heuristics.

3.1.1 Link Analysis Heuristics

The relationship between the web pages is evaluated by the Link analysis heuristic technique. There are three features taken into account based on the Link analysis heuristics. They are NIL iteration, Solitary and numerous iterations.

(i) Nil Iteration (H1)

If a webpage does not travel to the same page again, it indicates Nil iteration. There is no hyperlink pointing to the same page a second time.

(ii) Solitary Iteration (H2)

If a website travels to the same page again, it is indicated as a solitary iteration. Meeting the solitary iteration heuristic might not always indicate that a site is involved in phishing.

(iii) Numerous Iterations (H3)

If a website travels to the same page more than once, it is indicated as numerous iterations. If the requests and the corresponding responses result in the formation of more than one iteration, the numerous-iteration feature is satisfied.

These are the three features that are identified under the Link analysis heuristics. Each feature has been implemented and tested using test data.

3.1.2 Decision based Heuristics

Decision-based heuristics are evaluated based on the decisions needed when handling the forms in the web pages. There are three features that are considered under this Heuristics technique. They are the highest number of form postings, highest number of mistakes, and the feedback with the given input.

(i) Highest Number of Form Postings (H4)

When a random input is provided to a form, legitimate websites are expected to have a particular number of form submissions. However, phishing websites accept any number of form submissions. In the proposed system, the highest number of form submissions considered was three. If the website exceeds three, the website will be termed as a suspected phishing site.

(ii) Highest Number of Mistakes (H5)

Legitimate websites do not encourage random inputs. Thus, when random inputs are provided, legitimate websites generate an error message, whereas the phishing websites do not. Moreover, legitimate websites prevent the user from providing random inputs. In this system, the highest number of mistakes considered was three. If the user received more than three mistakes, the website will be considered as a phishing site.

(iii) Feedback with the given Input (H6)

This feature is fulfilled if the response to a form submission contains parts of the random inputs that have been supplied. A legitimate website often acknowledges a user after a login or registration with the name, user id, or email. Alternately, a phishing website does not generate a reply page that contains the supplied inputs.

These are the three features identified under the Decision-based heuristics. It is concluded that the legitimate websites will not encourage random number of inputs. If the website accepted the maximum number of form submissions, that website will be considered suspect. Legitimate sites provide an error message for random inputs and most of the phishing websites do not acknowledge the user after a successful login.

3.1.3 Content-based Heuristics

Content-based heuristics are evaluated, based on the contents which are available in the web pages. There are two features that are considered under this Heuristics technique. They are the websites containing images and text only.

(i) Linking Images (H7)

The website consists of images which has a link to another page.

(ii) Text only (H8)

The website consists of only text messages.

These are the two features identified under the category of content-based heuristics.

Thus, these eight features falling under the three heuristics types are taken into consideration in our study. It is observed that, using those three heuristics, a user is easily able to distinguish all phishing websites from legitimate websites. Moreover, a specific combination of Heuristics can be identified.

Thus, these eight features falling under the three heuristics types are taken into consideration in our study. It is observed that, using those three heuristics, a user is easily able to distinguish all phishing websites from legitimate websites. Moreover, a specific combination of Heuristics can be identified.

3.2. Feature Combinations

The heuristic combinations for most of the phishing websites satisfy the features (Nil Iteration (H1), Highest number of form postings (H4), Linking Images(H7)), and some of the URLs satisfy Numerous iteration (H3) and Highest number of form postings (H4). IT was therefore concluded, that many of the phishing websites do not have iterations and have only a single page. That single page consists of forms, and it satisfied the 'Highest number of form posting' feature (H4). Only phishing websites do not have any form submission restrictions.

For the legitimate websites, there is no specific combination to be satisfied. Though 'nil iteration (H1)' is satisfied in legitimate websites, 'acknowledging the user (H6)' is also satisfied. 'Acknowledging the user (H6)' mostly happens only with legitimate websites.

4. Experimental Results and Evaluation

To classify a website as a phishing site, the webpage is fetched, using a script and stored in memory. The HTML DOM objects are parsed for the information such as, presence of forms, number of iterations, form submission, number of mistakes and links. The response URLs of the forms are identified and also fetched. Random data is sent as POST/GET data to the forms, according to the form's method attribute and the response is analyzed. The data sent to the form are randomly generated. The script is then tested with number of websites, and the results are stored in a MySQL database.

4.1 Evaluation of False Negative Rate

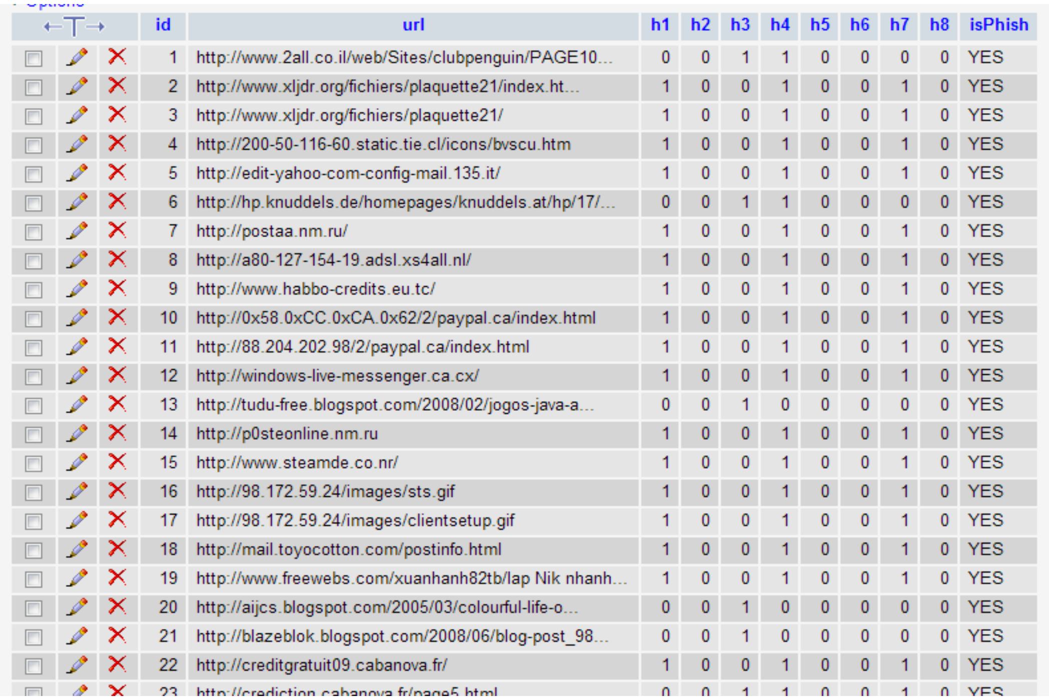
Evaluation is based on the phishing websites and legitimate websites. Two experiments were conducted with two different testing data sets for phishing and legitimate websites. The false negative and false positive rates are identified to find how well the proposed approach can detect the phishing and legitimate websites. A false negative rate indicates the rate at which a phishing web page is falsely identified as a legitimate web page.

Testing the reported URLs of the Phishtank and PhishLoad proves the effectiveness of this system. The testing dataset is built by collecting 1235 phishing URLs; 835 URLs were randomly taken from the Phishtank, and 400 were randomly taken from the PhishLoad. The different URLs are reported in the PhishTank and PhishLoad repository using different domain names. Figure 2 shows the evaluation result of the phishing websites. In a 1235 dataset, 1076 URLs satisfied the combinations of (Nil Iteration (H1), Highest number of form postings (H4), Linking Images(H7)) and the next highest combinations satisfied phishing was (Numerous iterations(H3) and Highest number of form postings (H4)). It is thus inferred that most of the phishing websites satisfy 'Nil iteration (H1)',' Highest number of form postings(H4)' and the 'form contains images' (H7). Some of the phishing websites satisfy 'multiple iterations' and 71 URLs satisfied these (Numerous iterations (H3) and 'Highest number of form postings' (H4)) combinations. This approach shows that the false negative rate is zero.

4.2 Evaluation of the False Positive Rate

The criteria used in this case is evaluating whether the approach can detect legitimate websites based on a specific heuristics combination. A new testing dataset is built by collecting 1000 legitimate URLs from Alexa.com. Figure 3 shows the evaluation results of the legitimate websites. In the legitimate website set, there is no specified set of combinations that satisfies the 'legitimate website' criteria. The system shows that the false positive rate is 0.03%.

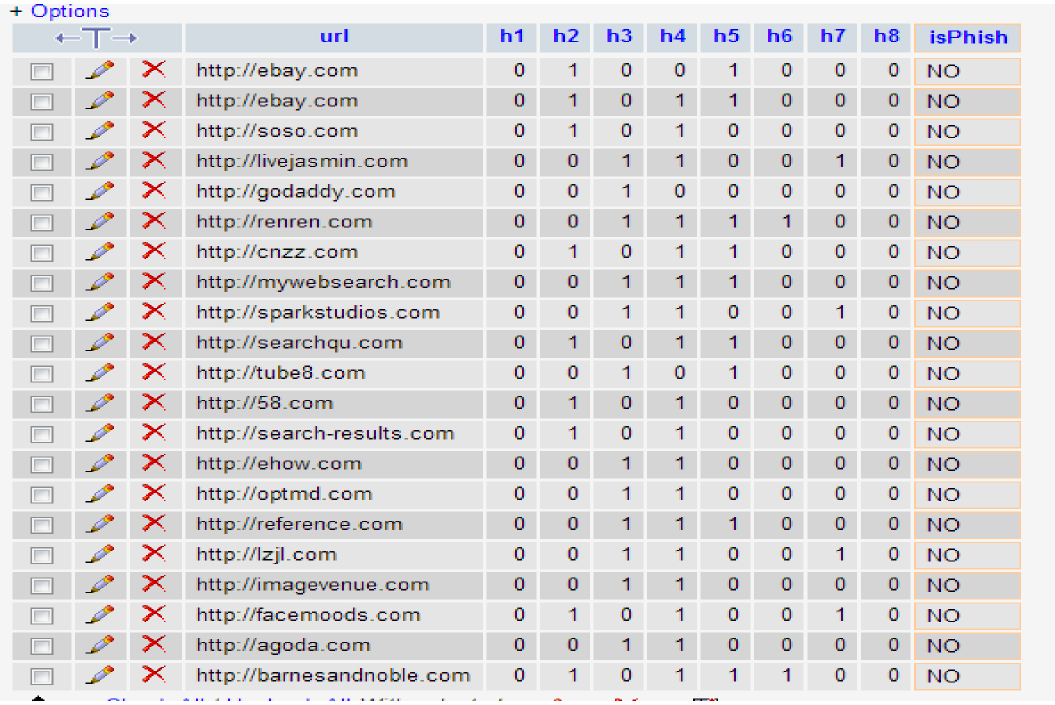
The results are classified using supervised learning algorithms to prove the correctness and to estimate the accuracy of the result. Supervised learning is the machine learning task of gathering a function from supervised training data.



+ Options

←T→			id	url	h1	h2	h3	h4	h5	h6	h7	h8	isPhish
□	✎	✕	1	http://www.2all.co.il/web/Sites/clubpenguin/PAGE10...	0	0	1	1	0	0	0	0	YES
□	✎	✕	2	http://www.xljdr.org/fichiers/plaquette21/index.ht...	1	0	0	1	0	0	1	0	YES
□	✎	✕	3	http://www.xljdr.org/fichiers/plaquette21/	1	0	0	1	0	0	1	0	YES
□	✎	✕	4	http://200-50-116-60.static.tie.cl/icons/bvscu.htm	1	0	0	1	0	0	1	0	YES
□	✎	✕	5	http://edit-yahoo-com-config-mail.135.it/	1	0	0	1	0	0	1	0	YES
□	✎	✕	6	http://hp.knuddels.de/homepages/knuddels.at/hp/17/...	0	0	1	1	0	0	0	0	YES
□	✎	✕	7	http://postaa.nm.ru/	1	0	0	1	0	0	1	0	YES
□	✎	✕	8	http://a80-127-154-19.adsl.xs4all.nl/	1	0	0	1	0	0	1	0	YES
□	✎	✕	9	http://www.habbo-credits.eu.tc/	1	0	0	1	0	0	1	0	YES
□	✎	✕	10	http://0x58.0xCC.0xCA.0x62/2/paypal.ca/index.html	1	0	0	1	0	0	1	0	YES
□	✎	✕	11	http://88.204.202.98/2/paypal.ca/index.html	1	0	0	1	0	0	1	0	YES
□	✎	✕	12	http://windows-live-messenger.ca.cx/	1	0	0	1	0	0	1	0	YES
□	✎	✕	13	http://tudu-free.blogspot.com/2008/02/jogos-java-a...	0	0	1	0	0	0	0	0	YES
□	✎	✕	14	http://p0steonline.nm.ru	1	0	0	1	0	0	1	0	YES
□	✎	✕	15	http://www.steamde.co.nr/	1	0	0	1	0	0	1	0	YES
□	✎	✕	16	http://98.172.59.24/images/sts.gif	1	0	0	1	0	0	1	0	YES
□	✎	✕	17	http://98.172.59.24/images/clientsetup.gif	1	0	0	1	0	0	1	0	YES
□	✎	✕	18	http://mail.toyocotton.com/postinfo.html	1	0	0	1	0	0	1	0	YES
□	✎	✕	19	http://www.freewebs.com/xuanhanh82tb/lap Nik nhanh...	1	0	0	1	0	0	1	0	YES
□	✎	✕	20	http://aijcs.blogspot.com/2005/03/colourful-life-o...	0	0	1	0	0	0	0	0	YES
□	✎	✕	21	http://blazeblok.blogspot.com/2008/06/blog-post_98...	0	0	1	0	0	0	0	0	YES
□	✎	✕	22	http://creditgratuit09.cabanova.fr/	1	0	0	1	0	0	1	0	YES
□	✎	✕	23	http://crediction.cabanova.fr/page5.html	0	0	1	1	0	0	1	0	YES

Figure 2. Sample Phishing Websites



+ Options

←T→			url	h1	h2	h3	h4	h5	h6	h7	h8	isPhish
□	✎	✕	http://ebay.com	0	1	0	0	1	0	0	0	NO
□	✎	✕	http://ebay.com	0	1	0	1	1	0	0	0	NO
□	✎	✕	http://soso.com	0	1	0	1	0	0	0	0	NO
□	✎	✕	http://livejasmin.com	0	0	1	1	0	0	1	0	NO
□	✎	✕	http://godaddy.com	0	0	1	0	0	0	0	0	NO
□	✎	✕	http://renren.com	0	0	1	1	1	1	0	0	NO
□	✎	✕	http://cnzz.com	0	1	0	1	1	0	0	0	NO
□	✎	✕	http://mywebsearch.com	0	0	1	1	1	0	0	0	NO
□	✎	✕	http://sparkstudios.com	0	0	1	1	0	0	1	0	NO
□	✎	✕	http://searchqu.com	0	1	0	1	1	0	0	0	NO
□	✎	✕	http://tube8.com	0	0	1	0	1	0	0	0	NO
□	✎	✕	http://58.com	0	1	0	1	0	0	0	0	NO
□	✎	✕	http://search-results.com	0	1	0	1	0	0	0	0	NO
□	✎	✕	http://ehow.com	0	0	1	1	0	0	0	0	NO
□	✎	✕	http://optmd.com	0	0	1	1	0	0	0	0	NO
□	✎	✕	http://reference.com	0	0	1	1	1	0	0	0	NO
□	✎	✕	http://lzjl.com	0	0	1	1	0	0	1	0	NO
□	✎	✕	http://imagevenue.com	0	0	1	1	0	0	0	0	NO
□	✎	✕	http://facemoods.com	0	1	0	1	0	0	1	0	NO
□	✎	✕	http://agoda.com	0	0	1	1	0	0	0	0	NO
□	✎	✕	http://barnesandnoble.com	0	1	0	1	1	1	0	0	NO

Figure 3. Sample Legitimate Websites

The training data consists of a set of training examples. Each example consists of an input object and a desired output value, and they form a pair. Five classification algorithms namely, Decision Tree Induction (DT), Multilayer Perceptron (MLP), Naïve Bayes (NB), Bayesian Network (BN) and Radial Basis Function (RBF)are used for classifying the website data. The classification algorithms are implemented and trained using WEKA (Waikato Environment for Knowledge Analysis). The Weka 3.4.4, a GUI-based workbench, is a collection of machine learning algorithms and data pre-processing tools. The robustness of the classifiers is evaluated, using a 10–fold cross validation. The phishing website is assessed by the primary performance measure – 'Predicted Accuracy'. The performances of the trained models are evaluated, based on the prediction accuracy and the build (training) time. The 10-fold cross validation results of the five classifiers are summarized in Table 1.

Table 1 Performance Comparison between Classifiers

Criteria for Evaluation	Supervised Learning algorithm				
	DT	**MLP**	**NB**	**BN**	**RBF**
Kappa Statistic	0.96	0.97	0.96	0.96	0.96
Mean Absolute Error	0.0101	0.0151	0.0258	0.0246	0.0101
Relative Absolute Error (%)	2.9369	4.5651	3.9367	5.4326	7.3745
Root Relative Squared Error (%)	20.7161	24.3131	35.5463	23.4532	25.6743
Correctly classified instances (%)	99.4924	98.5634	98.3212	97.3433	96.2415
Incorrectly classified instances (%)	0.5076	1.4366	1.6788	2.6567	3.7585
Precision (%)	99	98	97.4	96.4	95.4
Recall (%)	99.5	98.3	97.1	98.9	95.6
F Measure (%)	99.2	97.6	96.5	98.3	97.1
ROC Area(%)	98.8	98.6	95.4	91.75	95.4
Time taken to build model (Sec)	0.15	0.95	0.45	0.83	0.98

The performance evaluation based on kappa statistics, mean absolute error, relative absolute error and root relative squared error are shown in Table 1. Kappa, which is the possibility corrected measure of agreement, is calculated by taking the agreement expected by chance, away from the observed agreement, and dividing it by the maximum possible agreement. In the test set, Weka obtains a distribution for each instance, which is matched against the expected distribution. The absolute error is calculated for each class label, and the sum of these, gives the absolute error of each instance. The sum of all the instances and their absolute error instances, when divided by the number of instances in the test set, gives the mean absolute error. The difference between the forecast and its observed values gives the RMSE (Root Mean Squared Error). Here, the values are squared and then averaged over the sample value, which is then square rooted to get the mean value. Squaring the errors before taking the average gives high weightage to large errors. The Receiver Operating Characteristic (ROC) report shows the average performance of the experiment.

The comparison results between the classifiers for Precision, Recall and F Measure are shown in Figure 4. It clearly shows that the accuracy is higher when classified using Decision-Tree Induction algorithm that uses these set of features. Moreover, it is found that the time taken to build the PhishDetector and the precision accuracy are high when using Decision-Tree Induction, compared to the other four algorithms, as shown in Figure 5. However, the Precision, Recall and F Measure of all five classification algorithms (DT, MLP, NB, BN and RBF) are more than 95%. For Decision tree Induction (DT) the Precision, Recall and F measure are 99%, 99.5% and 99.2% respectively. Thus, it is observed that the overall classification accuracies using the five classification algorithms, is also superior in performance. Figure 6 shows the Receiver Operating Characteristic (ROC) curve, which is a plot of the true positive rate against the false positive rate for the five classifiers. It also clearly shows that the accuracy is higher when classified using the Decision-Tree Induction algorithm using these set of features.

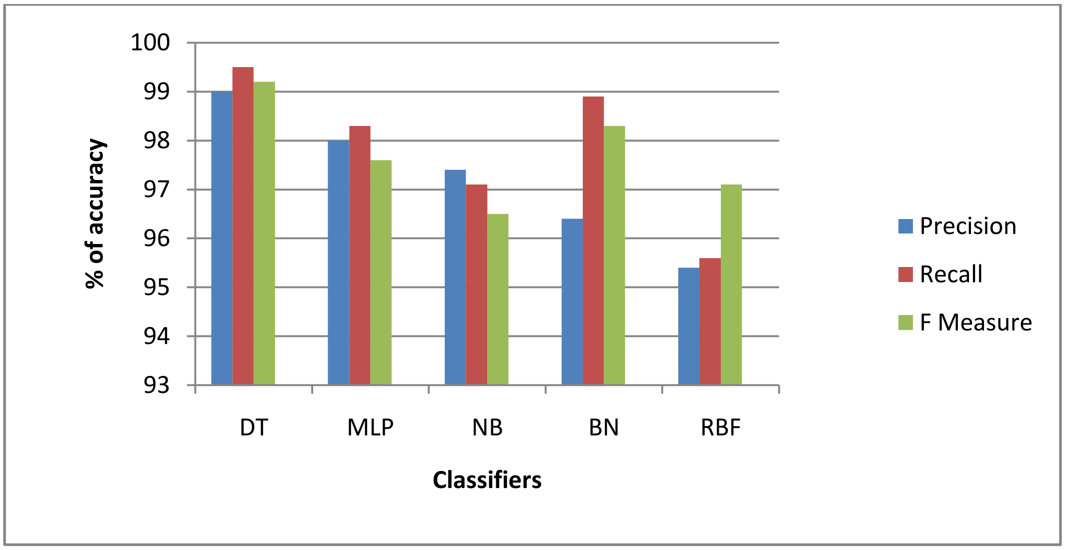


Figure 4. Comparison Graph between Classifiers

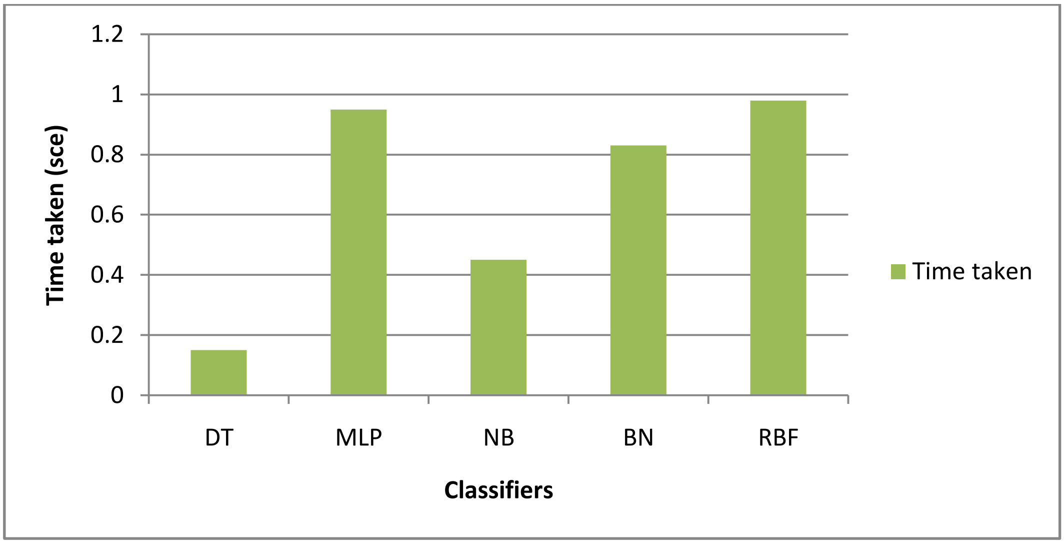


Figure 5. Building Time between the Classifiers

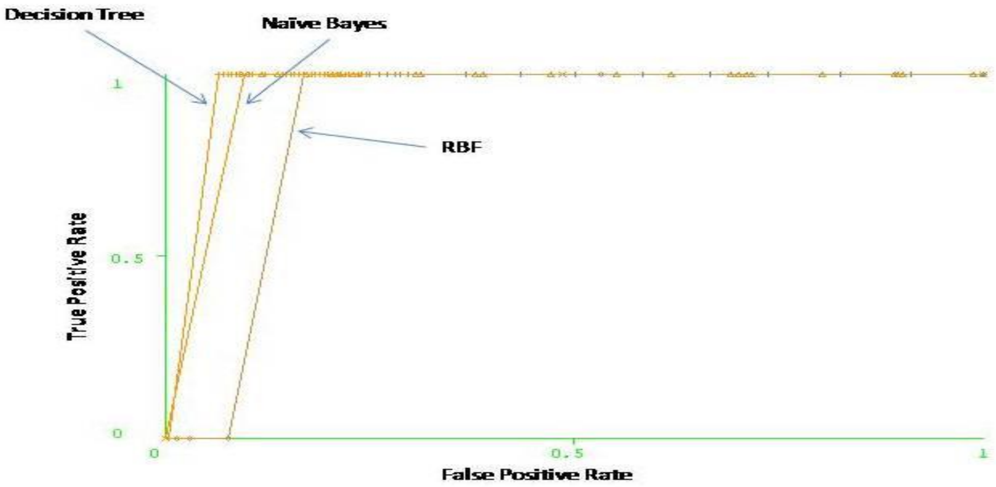


Figure 6 Receiver Operating Characteristic against True Positive and False Positive Rate

4.3 Performance Comparison with the Existing Techniques

The behavioral model is compared with the existing techniques like CANTINA [12], Anomaly method [5], Webpage detector [18], and the tools SpoofGuard and NetCraft. SpoofGuard and NetCraftwere selected for this comparisonamong the many tools available, because SpoofGuard had the highest True Positive, and NetCraft was one of the best toolbars. The performance comparison is based on the True Positive and False Positive rates as shown in Table 2.

Table 2 Performance Comparison with the existing Techniques

Methods	**True Positive (%)**	**False Positive (%)**
CANTINA	89	1
Anomaly	94	14
Webpage detector	97	4
SpoofGuard	91	48
NetCraft	97	0
Behavioral Model (PhishDetector)	100	0.03

The proposed behavioral model approach precisely detects the phishing pages, while it continues to have a low false positive rate. The anomaly result and the CANTINA result are collected from the corresponding empirical studies [5] [12]. Figures 7 and 8 show the True-Positive and False-Positive values of the PhishDetector in comparison with existing techniques. The next metric used to evaluate the performance is Precision. Table 3 shows the precision values of all the existing methods with the Behavior-based approach. The comparison graph based on the resultant accuracy is shown in Figure 9. Based on the results achieved, it has been inferred that the use of feature combination findings help in achieving better results using the 'Decision Tree Induction' classifier.

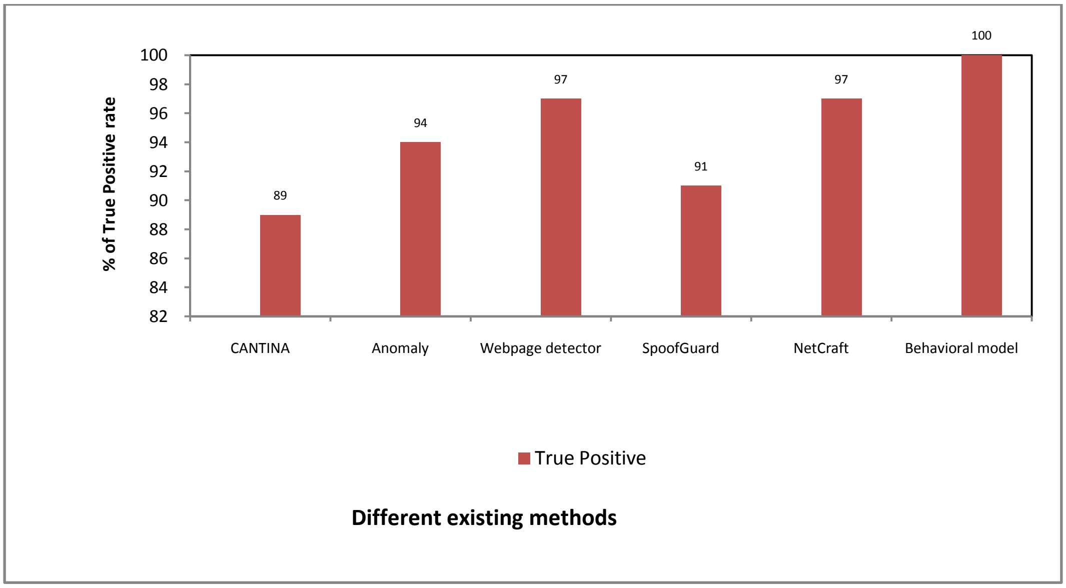


Figure 7. Comparison of PhishDetector with the Existing Techniques based on True Positive

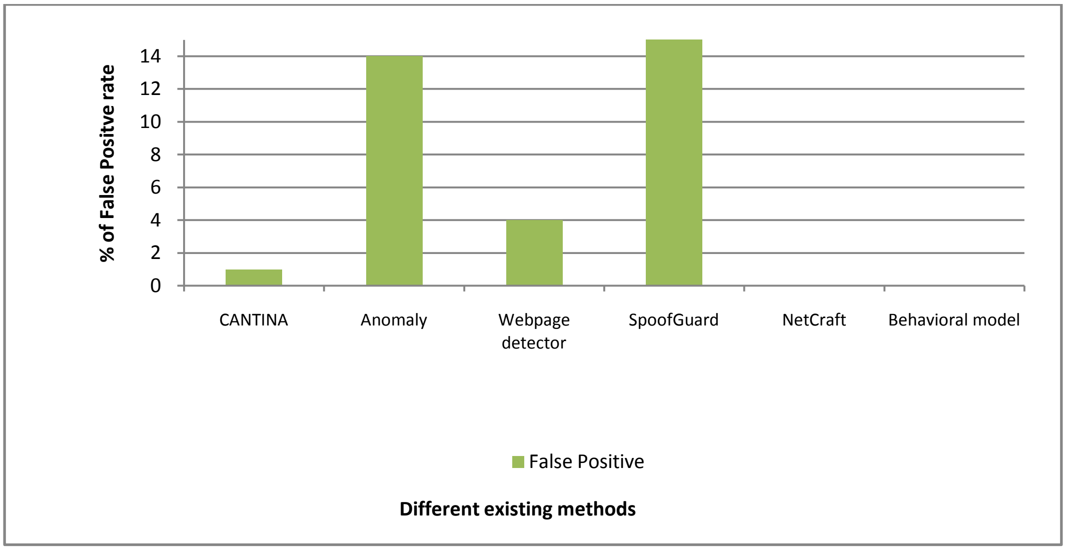


Figure 8. Comparison of PhishDetector with the existing techniques based on False Positive

The next metric used to evaluate the performance is Precision. Table 3 shows the precision values of all the existing methods with the Behavior-based approach. The comparison graph based on the resultant accuracy is shown in figure 9. Based on the results achieved, it has been inferred that the use of feature combination findings help in achieving better results using the 'Decision Tree Induction' classifier.

Table 3 Performance Comparison with existing Methods

Methods	Precision
CANTINA	88.88
Anomaly	87.66
Webpage detector	96.75
SpoofGuard	65.45
NetCraft	99.99
Behavioral Model (PhishDetector)	99.99

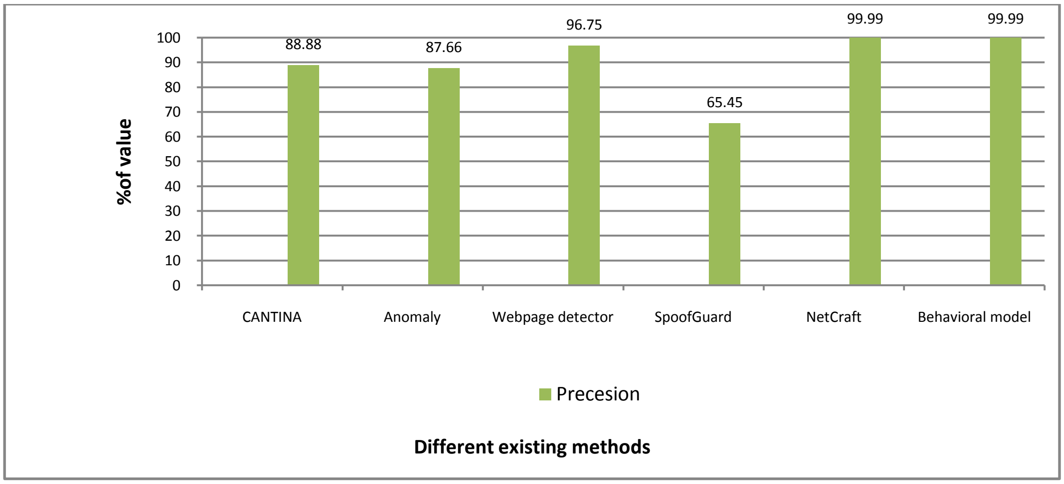


Figure 9. Comparison of PhishDetectorwith Existing Methods based on Accuracy

From the results achieved, it has been found that the proposed PhishDetector performs admirably well to a great degree of satisfaction, achieving a 0% false negative, which is actually more efficient than other existing approaches in this domain.

5. Conclusion and Future Work

If you Phishing is an ever-increasing concern for website users. A majority of the anti-phishing tools used today are geared towards the user, with only a small attempt being made to computerize the task of anti-phishing experts, who normally only manually verify an accounted website. We have developed eight different features and their different combinations which could aid these experts. We have also implemented the PhishDetector tool in the .NET framework. The tool has been evaluated using 1235 phishing URLs and 1000 legitimate websites of different combinations. The empirical results show that the false negative rate is 0% and false positive rate is 0.03%.The overall classification accuracy achieved for DT, MLP, NB, BN and RBF in sequential order is99%, 98%, 97.4%, 96.4% and 95.4%.

As an extension of this work, further development is planned to include Heuristics for E-mail based phishing. The current version of the PhishDetector does not handle E-mail phishing.

References

D. Geer, Security technologies go phishing, Computer Archive .38 (6), 18-21 (2005)

D.Irani, S.Webb, J,Giffin, C.Pu. Evolutionary study of phishing. In: Proc of the 3rd Anti-Phishing Working Group eCrime Researchers Summit, Atlanta, Georgia, October 1-10, (2008)

I. Fette, N. Sadesh, A. Tomasic, Learning to detect phishing emails, in: Proc. Of the 16th Intl. Conf. on World Wide Web, Banff, Alberta, Canada,649-656 (2007)

J. Kang, D.Lee, advanced White list approach for preventing access to phishing sites, in:Proc.of the Intl.Conf.on Convergence Information Technology, korea, 491-496 (2007)

Y.Pan, X. Ding, Anomaly-based web phishing page detection, in: Proceedings of the 22nd Annual Computer Security Applications Conference, Miami, Florida, December 381-392 (2006)

C.Xiang.J.Hong. A hybrid phish detection approach by identity discovery and keywords retrieval, in: Proceedings of the 18thIntl.Conference on World Wide web, Madrid, Spain, 571-580 (2009)

N. Chou, R. Ledesma, Y.Teraguchi, D.Boneh, J.Mitchell, Client side defense against web-based identity theft, in: Proc of the 11th Annual Network and Distributed Security Symposium, NDSS'04, San diego, CA, February (2004)

L.Wenyin, N. Fang, X. Quan, B. Qiu, G. Liu, Discovering phishing target based on semantic link network, Future Generation Computer Systems. 26 (3) 381-388, (2010)

B. Nelson, M. Barreno, F.J. Chi, A.D. Joseph, B.I.P. Rubinstein, U.saini, C.Sutton, J.D. Tygar, 9nd K. Xia. Exploiting machine learning to subvert your spam filter. In LEET: Proceedings of the IstUsenix Workshop on Large-Scale exploits and Emergent Threats, Pages 1-9, Berkeley,

Ye Cao, Weili Han, Yueran Le, Anti-phishing Based on Automated Individual White-List , Copyright 2008 ACM doi: 978-1-60558-294-8/08/10

W.Liu, G. Huang, A. Fu, An antiphishing strategy based on visual similarity assessment, IEEE Internet Computing. 10 (2), 58-65 (2006)

Y.Zhang, J. hong, L.Cranor, CANTINA: A content based approach detecting phishing websites, in: Proc.of the 16thIntl.Conf.on World wide Web, Banff, Alberta, 639-648 (2007)

Xin (Robert) Luo, wei Zhang, Stephen Burd, Alessandro seazzu, Investigating phishing victimization with the heuristic-Systematic Model: A theoretical framework and an exploration, Computers and security, 1-11 (2013)

HossainShahriar, Mohamed Zulkernine, Trustworthiness testing of phishing websites: A behavior model-based approach, Future Generation Computer Systems. 28, 1258-1271 (2012) www.w3schools.com/dom/dom_parser.asp

www.Phishtank.com

Crypto.stanford.edu/SpoofGuard/download.html

Mingxing He, Shi-Jinn Horng, Pingzhi Fan, Muhammad Khurram Khan, Ray-Shine Run, Jui-Lin Lai, Rong-Jian Chen, and Adi Sutanto, An efficient phishing webpage detector. 38(10), 12018-12027 (2011)

www.kaspersky.co.in

www.netcraft.com