The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE’13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.9 e-ISSN: 2251-7545

Phishing Detection Plug-In Toolbar Using Intelligent Fuzzy-Classification Mining Techniques

Maher Aburrous
Software Engineering
ALHOSN University
Abu Dhabi, UAE
e-mail: m.aburrous@alhosnu.ae

Adel Khelifi
Software Engineering
ALHOSN University
Abu Dhabi, UAE
e-mail: a.khelifi@alhosnu.ae

Abstract—**Detecting phishing website is a complex task which requires significant expert knowledge and experience. So far, various solutions have been proposed and developed to address these problems. Most of these approaches are not able to make a decision dynamically on whether the site is in fact phished, giving rise to a large number of false positives. In this paper we have investigated and developed the application of an open source intelligent fuzzy-based classification system for e-banking phishing website detection. The main goal of the proposed system is to provide protection to users from phishers deception schemes, giving them the ability to detect the legitimacy of the websites. The proposed intelligent phishing detection system employed Fuzzy Logic (FL) model with classification mining algorithms. The approach combined the capabilities of fuzzy reasoning in measuring imprecise and dynamic phishing features, with the capability to classify the phishing fuzzy rules. The proposed intelligent phishing website detection system was developed, tested and validated by incorporating the scheme as a web based plug-in phishing toolbar. The results obtained are promising and showed that our intelligent fuzzy based classification detection system can provide an effective help for real-time phishing website detection. The toolbar successfully recognized and detected approximately 86% of the phishing websites selected from our test data set, avoiding many miss-classified websites and false phishing alarms.**

***Keywords-** phishing website detection; fuzzy logic; data mining; classification; e-banking security; intelligent plug-in toolbar*

I. Introduction

Phishing website is a very complicated and complex issue to understand and to analyze, since it is a combination of technical and social dynamics for which there is no known single silver bullet to solve it entirely. Many users believe that using on-line banking increases the likelihood that they will become victims of phishing websites and identity theft, even though on-line banking provides more secure identity protection than paper- and mail-based systems.

The most harmful effect is that it will create “trust crises”. The trust will be eroded gradually without effective countermeasures to deal with the fraud, and everyone participating in network transactions will be harmed in the end. Trust is one of the most important determinants of successful e-banking [1]. Many researchers have argued that trust is essential for understanding interpersonal behaviour and is relevant to e-banking. Trust is not merely a short-term issue, but also the most significant long-term barrier to realizing the potential of BtoC e-commerce [2]. Falling victim to phishing websites could steal a customer’s proprietary information such as their account information and passwords, trade secrets, or other intellectual assets. Theft of a customer’s confidential information could have a disastrous effect on the companies or banks using electronic technology and could damage the trust between them and their clients. Even in developed countries, many people are worried that their credit card details will be misused or hacked into, and are concerned about on-line fraud, such as phishing websites that offer imaginary services or items. Despite the great quantity of applications available for phishing website detection, there are only a few solutions that utilise machine learning mining techniques in detecting phishing websites. Moreover, most of these proposed and already implemented solutions are impractical, inaccurate and suffer from unacceptable levels of false positives or miss detection [3][4].

The motivation behind this research paper is to create a resilient and effective intelligent model to detect phishing websites and to discover whether phishing activity is taking place or not, in order to prevent all users from being deceived or hacked.

This research investigates intelligent phishing website detection system, based on an artificial intelligence (AI) supervised machine learning approach. The technique uses fuzzy logic with simple data mining associative classification techniques and algorithms to process the phishing data features and patterns, for extracting classification rules into the data miner. The proposed phishing website system combines these techniques together to automate the fuzzy rules, produced by using the extracted classification rules to be implemented inside the fuzzy inference engine. These fuzzy rules allow us to construct if-then rules, which reflect the relations between the different phishing characteristics and features and their association with each other, to be used for the final phishing website detection rate. we believe that a hybrid system which combines and integrates fuzzy logic with data mining technique, using variation of associative classification algorithms (CBA, JRip, PART, PRISM, C4.5) implemented into the Data Miner, allows for valuable phishing feature extraction and rule processing, providing efficient techniques for classifying and indentifying phishing website with low false positive and false negative detection rate. This new mechanism reduces the need for human intervention and enhances the performance and the precision of detecting phishing websites rate.

We designed a plug-in phishing website detection toolbar for testing and validation using our integrated mining classification fuzzy model to show and prove its feasibility, reliability and accuracy. The implementation was programmed using Java language, and it successfully recognized and detected approximately 86% of the phishing websites selected from our test data subset, avoiding many miss-classified websites and false phishing alarms. Further, we show from this practical plug-in toolbar implementation that data mining classification fuzzy-based solutions are actually quite effective in protecting users against phishing websites attacks and improving existing anti-phishing applications.

II. Literature Review

Anti-phishing tools provide consumers with a dynamic system of warning and protection against potential phishing attacks, and they also defend the brands of legitimate ISPs and web commerce site developers from being "spoofed" to propagate scams. Of course, the most important role of an anti-phishing tool is to identify phishing websites in a very accurate way and within an acceptable timescale. Some of these tools provide binary indicators which show whether that site is phishing or not, and that can be implemented by using coloured indicators (green represents a legitimate site, and red represents a positively-identified phishing site). Other tools use a ternary system which means that the site can be phishing, legitimate, or unknown (suspicious), and that can also be implemented by using coloured indicators (green represents a legitimate site, red represents a positively-identified phishing site and a yellow or gray indicator represents an unknown or suspicious site).

Many proposed anti-phishing solutions use toolbars that show different types of security messages and warnings in the web browser's interface to help users detect phishing sites, such as Spoofguard [5], Trustbar [6], SpoofStick [7] and Netcraft [8] toolbars. Users are advised to look at the existing browser security indicators, e.g., the URL displayed in the address bar and the lock icon displayed in the status bar when a connection is SSL-protected. However, controlled user studies have shown that these security indicators are ineffective against high-quality phishing attacks for several reasons [9]:

First reason, warning indicators located in a peripheral area provide a much weaker signal than the centrally displayed web page and can be easily overwhelmed by convincing web content. Many users rely on the web content to decide if a site is authentic or phishing. *Second* reason, the security-related information shown by the indicators is not really needed for the user's current task. Since security is rarely a user's primary goal, users fail to pay continuous attention to the indicators. Making security a separate task that users are required to remember is not an effective solution. *Third* reason, sloppy but common web practices cause some users to rationalize the violation of the security rules that some indicators use to detect phishing attacks. For example, users are told to examine the hostname displayed in the address bar, to make sure that the hostname is the one they are expecting. But some legitimate websites use IP addresses instead of hostnames (*e.g.*, the Google cache) and some sites use domain names that are totally different from their brand names [10]. *Fourth* reason, some indicators deliver warnings without detailed, convincing explanations, which makes users think that the software is buggy and thus not treat the warning seriously. *Fifth* reason, although users do notice the system model displayed by the toolbar under phishing attacks, most of them do not have the expertise to correctly interpret it. For example, they cannot tell the difference between a lock icon displayed on a web page and the one displayed in the status bar. (e.g., amazon.com vs. amazon-department.com) are actually from the same organisation in the real world. *Finally*, security indicators tend to show that something is wrong and advise users not to proceed, but they do not suggest good alternatives. This may encourage users to risk submitting their information anyway, since they don't see any other way to accomplish their goal.

The phishing filter in IE8 is a toolbar approach with more features such as blocking the user's activity on a detected phishing site. The most popular and widely-deployed techniques, however, are based on the use of blacklists of phishing domains that the browser refuses to visit. For example, Microsoft has recently integrated a blacklist-based anti-phishing solution into its Internet Explorer (IE8). The browser queries lists of blacklisted and whitelisted domains from Microsoft servers and makes sure that the user is not accessing any phishing sites. Microsoft's solution is also known to use some heuristics to detect phishing symptoms in web pages [11]. Other browser-integrated anti-phishing tools include Google Safe Browsing [12] and McAfee SiteAdvisor [13]. Similar to the Microsoft IE 8 anti-phishing protection, Google Safe Browsing uses blacklists of phishing URLs to identify phishing sites. The disadvantage of the approach is that non-blacklisted phishing sites are not recognized. The success of a blacklist relies on massive amounts of data being collected at frequent intervals. In contrast, SiteAdvisor is a database-backed solution that is, however, mainly designed for protection against malware-based attacks (e.g., Spyware, Trojan horses, etc.). It includes automated crawlers that browse web sites, perform tests and create threat ratings for each visited site. Unfortunately, just like other blacklist or database-based solutions, SiteAdvisor cannot recognize new threats that are unknown and not in the database [14]. Verisign (2005) has also been providing a commercial anti-phishing service. The company is crawling millions of web pages to identify "clones" in order to detect phishing web sites. Furthermore, just like other large companies such as Microsoft, McAfee and Google, blacklists of phishing websites are maintained. Note that one problem with crawling and blacklists proposals could be that the anti-phishing organisations will find themselves in a race against the attackers. This problem is analogous to the problems faced by anti-virus and anti-spam companies. Obviously, there is always a window of vulnerability during which users are susceptible to attacks. Furthermore, listing approaches are only as effective as the quality of the lists that are maintained. [15] present a tool that tries to protect a client's identity and password information. They define client personality in terms of username, password and email address and introduce a function which provides clients with different personalities for the different servers they visit. [16] proposed inserting intelligent chip to sign as anti-phishing new fighting technique. [17] introduced new procedure by stemming software's flaws and improving vigilance with psychological defence, using different logon passwords and payment passwords. TrustedBrowser [18] uses a synchronized random coloured boundary to secure the path from users to their browser. The trusted status content is marked in the trusted window whereas the server content is shown in the distrusted window. Anti-Phish [19] compares the domains for the same sensitive information in web pages to the domains in the caches. That is, if it detects that confidential information such as a password is being entered into a form on a distrusted website, a warning is generated and the pending operation is cancelled. PhishHook [20] converts a web page to "normal form" through text, images and hyperlinks transformations. PwdHash [21], in contrast, creates domain-specific passwords that are rendered useless if they are submitted to another domain (e.g., a password for www.gmail.com will be different if submitted to www.attacker.com).

The limitation of browser-based schemes is that they require prior knowledge of the target site, which is unfortunately not always available.

III. Development Model and System Design Implementation

For our implementation of the fuzzy based classification mining model for phishing website detection, we have created our own intelligent phishing website detection toolbar as a plug-in for the Mozilla Firefox browser. Our intelligent toolbar helps the users to identify phishing websites effectively and dynamically. We used a standard version of JavaScript to extract the basic features of the website. To extract other sophisticated website features, like protocols (https), certificates (SSL) and DNS record, the desktop-based Java (J2SE 1.6) was used. For the application user interface we used standard browser based interface language XUL (XML User Interface Language).

We used the standard JavaScript to extract the website feature because we wanted to extend the application to all standard browsers in our future work. It will be easily adaptable to be integrated to all browsers which support JavaScript as well as its platform independent (Windows, Linux, Mac OS and UNIX) usability.

The proposed intelligent anti-phishing toolbar has the ability to extract all of our 27 phishing website features and patterns shown in Figure 1. It cross-check each extracted feature to validate the phishing vulnerability based on specified fuzzy sets to correspond them to related fuzzy variables (High, Moderate and Low) [26].

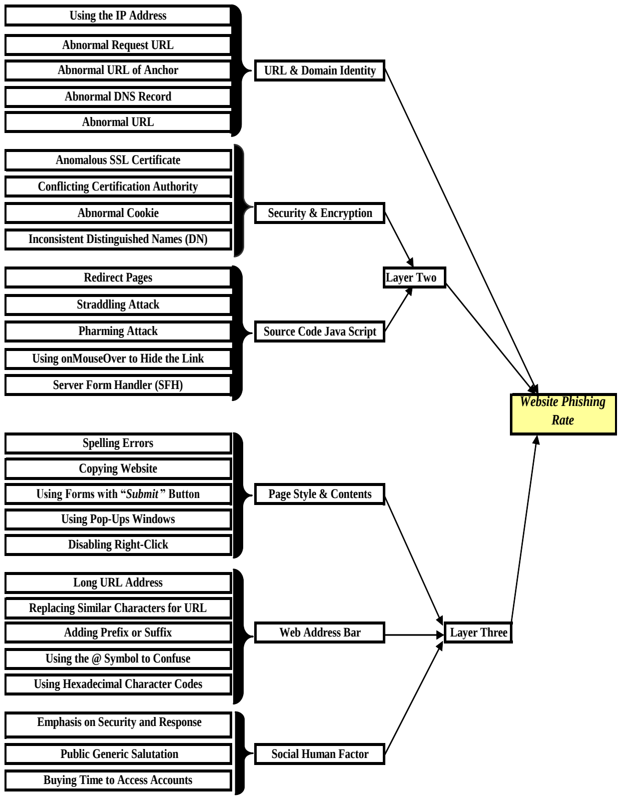


Figure 1. Layers of phishing website main features and criteria

The toolbar considers and fits each extracted phishing feature in its predetermined criteria and layer, based on risk significance and type. The system has defined six criteria (URL & Domain Identity, Security & Encryption, Source Code & Java Script, Page Style & Contents, Web Address Bar and Social Human Factor) and three layers (Layer One, Layer Two, Layer Three) [25]. We utilised the classification rules which were generated automatically from the associative classification data miner model to correlate each layer with its preceding layer output [27].

We used CBA application for generating AC rules, and WEKA application for implementing the different classification algorithms (JRIP, PART, PRISM, J48) for generating all classifier rules, which will all be integrated by the fuzzy inference engine as shown in Figure 2 to produce more accurate results for the final phishing website detection rate [28].

To define all associate rules for phishing features and patterns in every specific criterion at each particular layer, we adopted some rule pruning techniques based on the significance of the criteria and layer of phishing risk ranking and weight. We used the pruning technique to optimize the processing time for a prompt accurate result. For example, in layer one, if we got a high value as a fuzzy input variable for some phishing feature; we ignore checking other features on that layer. One of the most important conclusions that were generated from the data miner associative classification algorithms state that; finding only one high fuzzy input feature in any criteria is quite enough to make the outcome fraudulent or fake. Other rules also applied that; for any two moderate fuzzy input features, the whole criteria which contain those features will be considered as doubtful or uncertain.

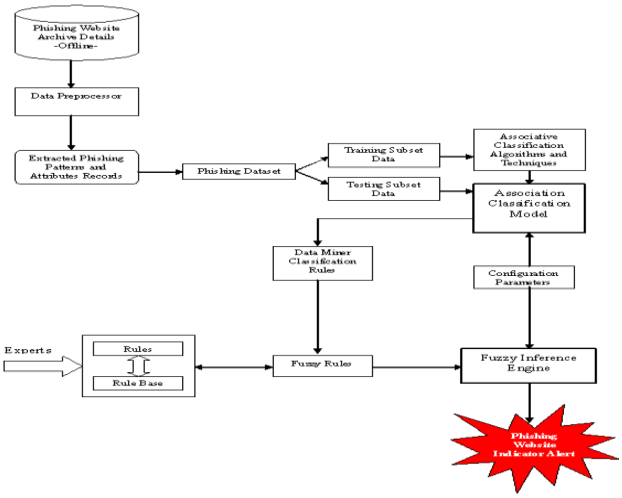

Phishing Website Archive Details -Offline-
Data Preprocessor
Extracted Phishing Patterns and Attributes Records
Phishing Dataset
Training Subset Data
Testing Subset Data
Associative Classification Algorithms and Techniques
Association Classification Model
Data Miner Classification Rules
Configuration Parameters
Experts
Rules
Rule Base
Fuzzy Rules
Fuzzy Inference Engine
Phishing Website Indicator Alert

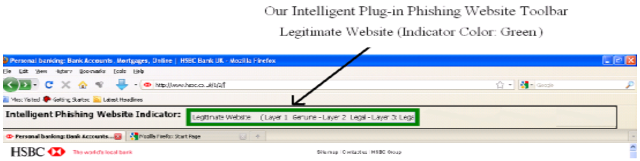
Figure 2. Intelligent classification fuzzy model for phishing detection

We used fuzzy-based classification mining approach with pruning technique to make the toolbar more effective and efficient than any other traditional phishing detection technique that uses black-listing or white-listing approach. The success of black-listing or white-listing depends on an extensive massive database which makes the response time much slower and impractical. Also this technique needs frequently-updated data, which makes it totally unreliable and not effective on 0 days attacks or spear attacks. Our technique outperforms the old existing techniques in terms of detection rate, response time, reliability and accuracy.

For our implementation, we have imported all the output of WEKA and CBA classification rules and saved the output in a CSV file. From this file, we have created a pool of classification rules to be integrated into our intelligent phishing detection toolbar through classification rule table. The advantage of integrating the rules table in our solution is to have the ability for our application to be flexible and dynamic. To introduce any new phishing classification rules, all we have to do is add the classification rules into the rule table, passing up the need for doing any kind of modification towards the application each time a new phishing classification rule is evolved. The defuzzification equation was implemented in our intelligent phishing detection toolbar to defuzzify the extracted fuzzy variables, having the role of fuzzy inference engine. Our implemented plug-in phishing detection toolbar managed to detect and identify approximately 86% of the phishing websites extracted from our test data subset, avoiding many misclassified websites and false phishing alarms.

IV. Plug-In Toolbar (Screen Shots & Open Source Code)

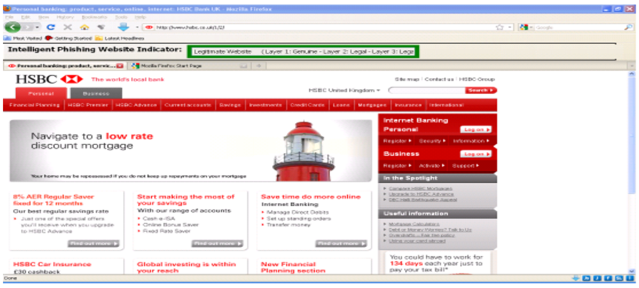
Figure 3 and Figure 4 shows screenshots of our intelligent plug-in phishing website detection toolbar for testing the legitimacy of the HSBC official e-banking website (www.hsbc.co.uk). Our intelligent toolbar checked all extracted 27 phishing features and patterns that can be found on this site. Then using the fuzzy-based classification rule mining approach adopted by our intelligent toolbar, all layered phishing features and patterns were associated and classified with each other for the final detection decision. Since the outputs of the three layers for that website were "genuine" and "legal", the final phishing detection rate was "Legitimate website" with the green colour indicator to make it more observable for users. We used the green colour for legitimate websites, red for phishing websites and yellow for suspicious websites.


Our Intelligent Plug-in Phishing Website Toolbar
Legitimate Website (Indicator Color: Green)

Personal banking: Bank Accounts, Mortgages, Online | HSBC Bank UK - Mozilla Firefox

Intelligent Phishing Website Indicator: Legitimate Website (Layer 1: Genuine - Layer 2: Legal - Layer 3: Lega

Figure 3. Our plug-in phishing detection toolbar

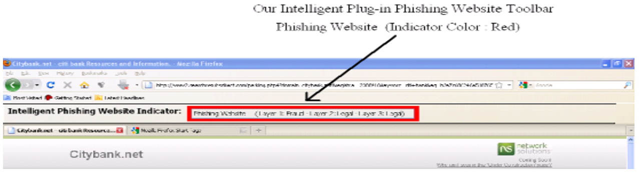

Personal banking: product, service, online, internet: HSBC Bank UK - Mozilla Firefox

Intelligent Phishing Website Indicator: Legitimate Website (Layer 1: Genuine - Layer 2: Legal - Layer 3: Legal

http://www.hsbc.co.uk/1/2/
HSBC The world's local bank
Navigate to a low rate discount mortgage
Internet Banking

Figure 4. Screen shot of legitimate website (hsbc.co.uk)

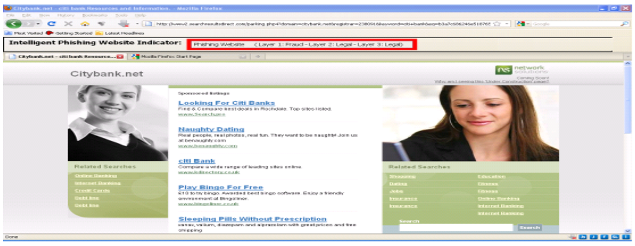
Figure 5 and Figure 6 shows screen shots for using our detection toolbar on a website for Citibank clients (Citybank.net). Since the outputs of the three layers for that website were mixed between "Fraud" for Layer one and "Legal" for Layer two and three, the final phishing detection rate was "Phishing Website" with a red colour indicator.


Our Intelligent Plug-in Phishing Website Toolbar
Phishing Website (Indicator Color : Red)

Citybank.net - citi bank Resources and Information. - Mozilla Firefox

Intelligent Phishing Website Indicator: Phishing Website (Layer 1: Fraud - Layer 2: Legal - Layer 3: Legal)

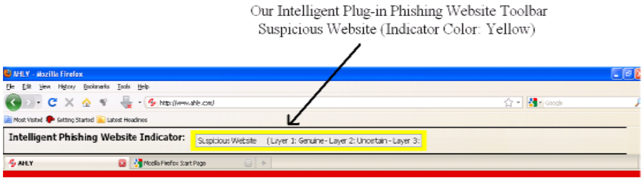
Figure 5. Our plug-in phishing detection toolbar (phishing website)



Intelligent Phishing Website Indicator: Phishing Website (Layer 1: Fraud - Layer 2: Legal - Layer 3: Legal)

Figure 6. Screen shot of phishing website (Citibank.net)

Figure 7 and Figure 8 shows screen shots using our detection toolbar on a website for Ahli bank clients (ahly.com). Since the outputs of the three layers for that website were mixed between "Genuine" for layer one and "Uncertain" for layer two and three, the final phishing detection rate was "Suspicious Website" with yellow colour indicator.

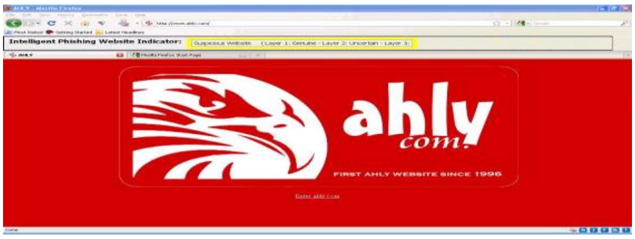


Our Intelligent Plug-in Phishing Website Toolbar
Suspicious Website (Indicator Color: Yellow)

Intelligent Phishing Website Indicator: Suspicious Website (Layer 1: Genuine - Layer 2: Uncertain - Layer 3:

AHLY - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.ahly.com/
Most Visited Getting Started Latest Headlines
AHLY
Mozilla Firefox Start Page

Figure 7. Our plug-in phishing detection toolbar (suspicious website)



Intelligent Phishing Website Indicator: Suspicious Website (Layer 1: Genuine - Layer 2: Uncertain - Layer 3:

Figure 8: Screen shot of phishing website (ahly.com)

We show now some important source code and pseudo code examples for extracting some of the phishing website features for our system implementation. This section also demonstrates how to validate the phishing features with our proposed phishing criteria and rate the fuzzy variable inputs accordingly.

- ***Extracting pop-up window phishing feature (Source code example)***

```
var popUpWindow = "Low";
        popUpCount = 0;
        var elems =
window._content.document.getElementsByTagName("script");
        if(elems){  for(i=0;i< elems.length; i++){
                                if(elems[i].innerHTML){
                                        var code =
elems[i].innerHTML;

        if(code.indexOf("window.open") > -1)
             {popUpCount++;
}}}}
        var toCheck = window._content.document.body.innerHTML;
                var findLock = toCheck.indexOf("window.open");
                        while(findLock > -1){
                var findLock = toCheck.indexOf("window.open",
findLock+1);
                popUpCount++; }
```

```
                if(popUpCount >= 0 && popUpCount < 2){
                popUpWindow = "Low";
        }else if(popUpCount >= 2 && popUpCount <8){
                popUpWindow = "Moderate";
        }else{
                popUpWindow = "High"; }
                if(popUpWindow == "High")
                return "Fraud";
        else if(popUpWindow == "Moderate" && formSubmit ==
"Moderate"){
                return "Doubtful"; }
```

Here we count how many times the pop-up window exists on the website. If it does not exist at all in the website, or there is only one pop-up window, then we give the fuzzy input variable "Low" value. If the number ranges from 2 to 7 pop-up windows then we give the fuzzy input variable "Moderate" value. Otherwise, we give it "High" fuzzy value.

- **Extracting redirect page phishing feature (Source code example)**

```
 var elems = window._content.document.getElementsByTagName("script");
        redirectCount = 0;
        usingRedirect = "Low";
        if(elems){
                for(i=0;i< elems.length; i++){
                        if(elems[i].innerHTML){
                         var code = elems[i].innerHTML;
                         var findLoc1 =
code.indexOf("window.location=\"");
                         if(findLoc1 > -1){
                                var findLoc2 =
code.indexOf("\"",findLoc1+1);
                                var toCheck =
code.substring(findLoc1,findLoc2);
                                url =
window.top.getBrowser().selectedBrowser.contentWindow.location.href;
                                domain =
url.split(/\/+/g)[1].replace('www.','');
                                pattern = "/"+domain+"/gi";
                                pattern = eval(pattern);
                                if(toCheck.match(pattern) ==
null)
                                        redirectCount++;  }
} }
        var elems =
window._content.document.getElementsByTagName("meta");
        if(elems){
                for(i=0; i< elems.length; i++){
                        toCheck = elems[i].content;
                        if(toCheck.indexOf("url=") > -1){
                                return redirectCount;
                                url =
window.top.getBrowser().selectedBrowser.contentWindow.location.href;
                                domain =
url.split(/\/+/g)[1].replace('www.','');
                                pattern = "/"+domain+"/gi";
                                pattern = eval(pattern);
                                if(toCheck.match(pattern) ==
null)
                                        redirectCount++; } }
}
                        if(redirectCount < 2)
                usingRedirect = "Low";
        else if(redirectCount >= 2 && redirectCount <= 4)
                usingRedirect = "Moderate";
        else
                usingRedirect = "High";
```

There are two ways to "Redirect Pages" from one site to another. The first is a script used to redirect with a syntax "window.location" and the other one is on the page where the <meta> refresh tag is used with a URL specified to the final targeted page. In this section of the code we considered both the possibility and count number of occurrences of these two techniques on a browsed page. To rate the Redirect Page feature as "High", the value of fuzzy input variable should have more than 4 occurrences. To rate Redirect Page feature as "Moderate", then value of fuzzy input variable should be between 2-4 occurrences. Finally, it will be rated as "Low" when the value is less than 2 occurrences.

- **Extracting abnormal URL anchor phishing feature (Pseudo Code Example)**

```
elems :- extract all window elements by the tag name a (Anchor);
url :- get the browsing URL address from the Location bar;
Domain :- get the Domain name part from the Whole URL without the "www" part;
Pattern :- make the pattern match using the extracted domain name;
notMatchedCountAnchor :- set the counter to 0;
abnormalURLRequestAnchor :- set the Fuzzy Variable to "Low" initially
Check if there is any anchor element
Do for every element
Check if the link URL does not match with the pattern
notMatchedCountAnchor :- increment the counter;  End
Calculate the percentage of mismatched found using the notMatchedCountAnchor counter and the number of Anchors in the page
notMatchedCountAnchorRatio :- (notMatchedCountAnchor/ total number of Anchor)*10
Check if notMatchedCountAnchorRatio is less then or equal to 20
abnormalURLRequestAnchor :- "Low";
Otherwise check if notMatchedCountAnchorRatio is in the range between 21 and 50
abnormalURLRequestAnchor :- " Moderate ";
Otherwise         abnormalURLRequestAnchor :- " High ";
```

To validate the "Abnormal URL Anchor" phishing feature, we have extracted all the anchor elements of the page. Then we have counted the number of anchors that were pointed at some other website other than the browsed domain name, and calculated the percentage of URLs that were pointed to some other websites. We have rated the fuzzy variable as "Low" If the percentage was less than 20%, "Moderate" if the percentage was between 21 -50 and "High" otherwise.

V. Implementation Constraints

We faced some implementation constraints regarding extracting and validating some of the 27 phishing website features. For example, validating the extracted spelling errors phishing feature was not 100% accurate since it included nouns which were not listed as dictionary words. These words would be considered as spelling errors resulting for about 25% error on spelling error detection.

Also we did not include WHOIS database query result with the validation process of phishing website features; because of the difficulties in extracting the data from WHOIS query result. That is the reason we could not validate the "Abnormal DNS Record" and "Abnormal Request URL" 100% accurately. The validation of these two features did not give the expected output for www.facebook.com, www.yahoo.com or any other website that uses a different valid and registered domain for images, scripts and other recourses. Nevertheless we are not facing this problem for e-Banking or e-Commerce sites, since they are very consistent in using their single domain to store every resource for security purposes.

VI. TESTING AND VALIDATION RESULTS

While there is no mature technology that defends against phishing web sites yet, there is currently no anti-phishing benchmark set of expectation or standardized set of data for phishing detection products evaluation. Most of the claims made by vendors of available products are based on proprietary test data and testing methodology. In this research paper, a test framework has been constructed which can evaluate a generic anti-phishing technology against the latest existing phishing sites. This framework has been used to evaluate the effectiveness of our intelligent plug-in phishing detection toolbar. We have selected the PhishTank data as the public benchmark for our phishing detection comparison. Details of this experimentation framework are presented below.

We tested our intelligent plug-in toolbar using a sample of 160 different e-banking website to confirm its validation and verification. The dataset sample was taken from the public benchmark Phishtank archive data [22], consisting of 80 phishing websites, 45 suspicious websites and 35 legitimate websites. Our toolbar managed to detect the phishing e-banking websites that were found in the testing sample with a very small miss-classification rate. The results indicate clearly the high precision of phishing classification with very small false positive and false negative rates, as shown in the confusion matrix Table 1.

TABLE I. Website Legitimacy Confusion Matrix

Decision Website Legitimacy	Legitimate	Suspicious	Phishing
Legitimate Website	**29**	**4**	**2**
Suspicious Website	**3**	**38**	**4**
Phishing Website	**3**	**8**	**69**

As shown in Table 1, there were just 6 legitimate websites miss-classified as suspicious or phishing websites, and only 11 phishing websites were miss-classified as legitimate or suspicious website.

These results demonstrate very clearly how effective and reliable detecting phishing website can be when applying an intelligent fuzzy-classification mining technique for website phishing detection. The enhancement to the final results was due to using an approach depending not only on the human expert knowledge alone, but also with intelligent approach, using fuzzy classification mining algorithms. When comparing our intelligent phishing detection plug-in toolbar with other well-known anti-phishing toolbars like Netcraft [8], Spoofstick [23] and Bitdefender [24] toolbars, we found that our toolbar outperformed the other detection toolbars regarding their accuracy, efficiency and speed. It managed to classify correctly approximately 86% of all tested phishing websites comparing to only 61% for Netcarft, 65% for Spoofstick toolbar and 73% for Bitdefender toolbar. We believe that the main reason for this observed enhancement and improvement on the phishing detection rate was due to our methodology on adopting a novel AI heuristic search on all phishing features and criteria that can be found on any website for intelligent phishing detection, beating all other anti-phishing toolbars which were depending only on using black-list and white-list databases in classifying phishing websites. Figure 9 shows the comparative analysis of all tested anti-phishing toolbars regarding their phishing accuracy rate.

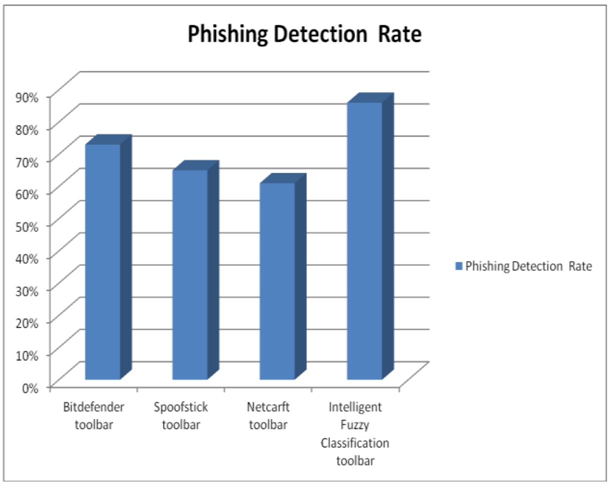


Figure 9. Phishing detection comparative analysis

VII. Conclusions And Future Work

A browser-based plug-in phishing detection toolbar has been implemented using an intelligent heuristic approach. The toolbar has extracted all phishing website features and patterns. Validation of the extracted features has been integrated into the solution to effectively identify phishing websites. An intelligent pruning technique has been used to increase the performance of the phishing detection rate. The intelligent phishing detection toolbar reduces the requirement of human knowledge intervention for of phishing website detection. Our toolbar has been provided as an alternative solution of depending only on the black-list or white-list approach by adopting a new fuzzy-based classification mining technique to detect phishing website. The results of our testing and validation shows that the proposed solution outperformed the existing phishing detection toolbars regarding its accuracy, efficiency and the speed of classifying and detecting phishing websites. It managed to classify correctly approximately 86% of all tested websites. The experimental results showed that both of the false-positive rate and the miss rate are reasonably low. A comparative performance analysis of the proposed model was presented in order to demonstrate the merits of capabilities through a set of experiments. It is noted that the proposed intelligent system offers better performance as compared to other existing tools and techniques.

During the implementation phase we faced some kind of complications regarding extracting and validating some of the phishing website features like spelling error extraction, validation of "Abnormal DNS Record" & "Abnormal Request URL" and shadow website copying. As a future work we will try to overcome and resolve these kind of challenges since it can be considered a major barrier for our intelligent solution to get its maximum performance and efficiency.

Also, we will try to make the phishing detection toolbar a desktop application to run as a background process for the independent phishing detection tool. Further, to utilize this application to increase the security awareness towards phishing website attacks to make it more effective, dynamic and interactive.

ACKNOWLEDGMENT

We would like to thank Mr. Nafi-Ul Karim for his valuable help regarding the technical programming part of our intelligent phishing website detection plug-in toolbar.

REFERENCES

[1] Suh, B., and Han, I. (2002) Effect of trust on customer acceptance of Internet banking. Electronic Commerce Research and Applications, Vol. 1, No. 3, (pp. 247-263).
[2] Gefen, D. (2002) Reflections on the Dimensions of Trust and Trustworthiness Among Online Consumers. ACM SIGMIS Database, Vol. 33, No. 3, (pp.38-53).
[3] Wu, M., Miller, R., and Garfinkel, S. (2006) Do Security Toolbars Actually Prevent Phishing Attacks?, CHI.
[4] Cranor, L., Egelman, S., Hong, J., and Zhang, Y. (2008) Phinding phish: An evaluation of antiphishing toolbars. In *Network & Distributed System Security (NDSS) Symposium,* CMU-CyLab-06-018.
[5] Chou, N., Ledesma, R., Teraguchi, Y., Boneh, D., and Mitchell, J. (2004) Client side defense against web-based identity theft. *In Proceeding of the 11th Annual Network and Distributed System Security Symposium (NDSS '04)*.
[6] Herzberg, A., and Gbara, A. (2004) Protecting Naive Web Users, Draft of July 18, 2004.
[7] Core Street: SpoofStick. (www.corestreet.com/spoofstick/)
[8] Netcraft. http://news.netcraft.com/
[9] Wu, M., Miller, R., and Little, G. (2006) Do Security Toolbars Actually Prevent Phishing Attacks?, CHI.
[10] Herzberg, A. (2005) Trustbar: "Re-establishing trust in the web", http://www.cs.biu.ac.il/˜herzbea/TrustBar/index.html, Access date [5/11/2006].
[11] Sharif, T. (2005) Phishing Filter in IE7, http://blogs.msdn.com/ie/archive/2005/09/09/463204.aspx, Access date [4/6/2007].
[12] Schneider, F., Provos, N., Moll, R., Chew, M., and Rakowski, B. (2007) Phishing Protection Design Documentation. http://wiki.mozilla.org/Phishing_Protection:_Design_Documentation, Access date [23/10/2007].
[13] *McAfee SiteAdvisor. http://home.**mcafee**.com/store/**siteadvisor**-live*
[14] Zhang, Y., Egelman, S., Cranor, L., and Hong, J. (2006) Phinding Phish: Evaluating Anti-Phishing Tools, Carnegie Mellon University, White Paper.
[15] Gabber, E., Gibbons, P., Kristol, D., Matias, Y., and Mayer, A. (1999) Consistent, yet anonymous, web access with LPWA. Communications of ACM, Vol. 42, No. 2, (pp. 42–47).
[16] Chandrasekaran, M. (2005) New Way to Avoid Phishing, Journal, Computer Software and Applications Conference, IEEE, Vol. 9, No. 7, (pp. 161-165).
[17] Chinchani, R., and Upadhyaya, S. (2005) Analysis of Phishing, Journal, World of Wireless, Vol. 7, No. 11, (pp. 70-73).
[18] Ye, Z., and Smith, S. (2005) Trusted paths for browsers. ACM Transactions on Information and System Security, Vol. 8, No. 2, (pp. 153–186).
[19] Kirda, E., and Kruegel, C. (2005) Protecting users against phishing attacks with antiphishing. *In Proceedings of the 29th Annual*

International Computer Software and Applications Conference (COMPSAC), (pp. 517–524).

[20] Stepp, M. (2005) Phishhook: A tool to detect and prevent phishing attacks. In DIMACS Workshop on Theft in E-Commerce: Content, Identity, and Service.

[21] Ross, B., Jackson, C., Miyake, N., Boneh, D., and Mitchell, J. (2005) Stronger Password Authentication Using Browser Extensions. *Proceedings of the 14th Usenix Security Symposium*.

[22] Phishtank, (2010),http://www.phishtank.com/phish_archive.php, Access date [28/4/2010].

[23] SpoofStick. http://www.spoofstick.com/

[24] Bitdefender. http://www.bitdefender.com

[25] Aburrous, M., Hossain, M.A., Dahal, K., Thabtah, F., "Modelling Intelligent Phishing Detection System for E-banking Using Fuzzy Data Mining"**,** *Proceeding of the International Conference on CyberWorlds (CW '09)*, pp. 265-272, IEEE Computer Society Press, Bradford, UK, 2009.

[26] Aburrous, M., Hossain, M.A., Dahal, K., Thabtah, F., "Intelligent phishing detection system for e-banking using fuzzy data mining", *Journal of Expert Systems with Applications*, Volume 37, Issue 12, Pages 7913-7921, DOI:10.1016/j.eswa.2010.04.044, Elsevier, NY, USA, December 2010.

[27] Aburrous, M., Hossain, M.A., Thabtah, F., Dahal, K., "Classification Techniques for predicting e-Banking Phishing Websites"**,** *Proceedings of the International Conference on Multimedia Computing and Information Technology (MCIT-2010)***,** pp. 9-12, IEEE**,** University of Sharjah, U.A.E, 2010.

[28] Aburrous, M., Hossain, M.A., Dahal, K., Thabtah, F.,"Predicting Phishing Websites using Classification Mining Techniques with Experimental Case Studies", *Proceedings of the 7th Int'l Conf. on Information Technology: New Generations ITNG 2010*,pp. 176-181, IEEE Computer Society Press, Las Vegas, USA, 2010.