Expert Systems With Applications 115 (2019) 300–313

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Review

Intelligent web-phishing detection and protection scheme using integrated features of Images, frames and text

M.A. Adebowale[a,*], K.T. Lwin[a], E. Sánchez[b], M.A. Hossain[a]

[a] Intelligent System Research Group, Anglia Ruskin Information Technology Institute, Anglia Ruskin University, Chelmsford, United Kingdom
[b] Department of Computing & Technology, Anglia Ruskin University, Chelmsford, United Kingdom

ARTICLE INFO

Article history:
Received 25 April 2018
Revised 29 July 2018
Accepted 29 July 2018
Available online 4 August 2018

Keywords:
Phishing
Support vector machine
ANFIS
Online transaction
Intelligent system

ABSTRACT

A phishing attack is one of the most significant problems faced by online users because of its enormous effect on the online activities performed. In recent years, phishing attacks continue to escalate in frequency, severity and impact. Several solutions, using various methodologies, have been proposed in the literature to counter the web-phishing threats. Notwithstanding, the existing technology cannot detect the new phishing attacks accurately due to the insufficient integration of features of the text, image and frame in the evaluation process. The use of related features of images, frames and text of legitimate and non-legitimate websites and associated artificial intelligence algorithms to develop an integrated method to address these together. This paper presents an Adaptive Neuro-Fuzzy Inference System (ANFIS) based robust scheme using the integrated features of the text, images and frames for web-phishing detection and protection. The proposed solution achieves 98.3% accuracies. To our best knowledge, this is the first work that considers the best-integrated text, image and frame feature based solution for phishing detection scheme.

1. Introduction

Phishing is a form of social engineering attack in which an attacker attempts to fraudulently retrieve sensitive user information by sending an email claiming to be a legitimately established organisation. They scam the user into giving confidential information that could be used for identity theft (Arachchilage, Love and Beznosov, 2016, Jakobsson & Myers, 2006). Commercial institutions and their end-users are regularly exposed to the threat of phishing attacks (Barraclough, Hossain, Tahir, Sexton, & Aslam, 2013). The danger is continuing to grow due to an increase in deception, impersonation, fraud and multiple online attacks. Most attacks are delivered by an email luring users to click a link embedded in the email that takes them to a malicious website. The attackers usually target end-users' financial information by claiming to be their bank, a utility company, HM Revenue and Customs or other government agencies to persuade the end-user to open the document attached to the email, which then targets sensitive information on their system (Office for National Statistics, 2017). The main reason why phishing attacks are still successful is the lack of awareness and computer literacy among Internet users mostly with regards to Internet safety (Deshmukh et al., 2017). According to a report by the Anti-Phishing Working Group (APWG) ,[1] the total number of phishing websites observed in the fourth quarter of 2017 was 296,208. The most targeted sector is the payment service with 41.99 % of phishing attacks, followed by software-as-a-service (SaaS)/webmail with 17.07 % and financial institution with 15.48 % (APWG, 2017).

Currently, the increase in the use of computer devices such as smartphones and tablets for accessing information on the Internet has more significance in financial crimes both regarding direct and indirect attacks (Fatt et al., 2014). That is to say, robbing a bank has changed into deceiving Internet banking users by stealing their identities and fraudulently using them to gain access to their Internet banking account and take their money (Moghimi & Varjani, 2016). According to a Financial Fraud Action[2] UK report for 2017, about £165 million lost to fraud related to Internet payment cards, remote banking and identity theft in 2017, and while this is 3.68% lower than the same period in 2016 (Financial Fraud

* Corresponding author.
E-mail addresses: moruf.adebowale@pgr.anglia.ac.uk (M.A. Adebowale), khin.lwin@anglia.ac.uk (K.T. Lwin), erika.sanchez@anglia.ac.uk (E. Sánchez), alamgir.hossain@anglia.ac.uk (M.A. Hossain).

https://doi.org/10.1016/j.eswa.2018.07.067
0957-4174/

[1] APWG report [Online] Available at: https://docs.apwg.org/reports/apwg_trends_report_q4_2017.pdf [Access 31 march 2018]

[2] Financial Fraud Action UK [Online] Available at: < https://www.financialfraudaction.org.uk/wp-content/uploads/2016/07/2017-half-year-fraud-update_September-17.pdf) [Access 15 December 2017]

Action, 2017). There is still a need to do more to protect Internet users from phishers who want to steal their confidential information for financial gain (Khadir, 2015).

Addressing this situation, Barraclough et al. (2013) proposed a robust model using neuro-fuzzy with few inputs. This model employs fuzzy logic in combination with a neural network. The purpose of using neuro-fuzzy is that it has universal approximation and the ability to use fuzzy *if…then* rules. Fuzzy logic performs well when dealing with the reasoning in high-level language information while neuro-fuzzy does well when dealing with raw data (Barraclough et al., 2013). Fuzzy logic is used to provide a mode of qualitative reasoning, which is closer to human decision making because it handles fuzziness and ambiguity by combining fuzzy fact and fuzzy relations. Hence, a neuro-fuzzy implementation helps a system to encode both unstructured and structured knowledge, while fuzzy rules enable the system to learn from examples (Stathacopoulou et al., 2005).

In this paper, a scheme that uses an intelligent ANFIS algorithm with a knowledge model and one input is proposed. The ANFIS is a network structure method that facilitates systematic computation of gradient vectors, it combines the least-squares and the gradient descent methods, and it utilises a useful fusion learning technique to derive the output error (Çakıt & Karwowski, 2016). ANFIS is a model that uses various features inputs selection, and it trains the data with the least-squares application (Jang, 1996). The proposed intelligent system will combine a neural network and fuzzy logic (Arachchilage, Love and Beznosov, 2016) with the capability of reasoning and learning through the knowledge model and external knowledge sources. This technique was chosen because it allows data learning by using the connectionist approach for computation, and therefore the exact rules are from the fuzzy inference point of view (Barraclough et al., 2013). The method in this study uses a table in which the features or data of a valid website are stored for reference purposes. The data will includes website images, text and frames features. A total of 35 features are extracted to model the ANFIS; 22 elements represent the structure of the text-based properties, eight features represent the frame-based properties, and five elements constitute the image-based resources of the website. The focus of the proposed intelligent phishing detection and protection scheme is to hinder website-based phishing attacks that aim to entice victims into giving out their confidential and sensitive information.

The main contributions of this study are the use of hybrid website features such as frames, images and text to improve on previous works (Barraclough et al., 2013) based on text only. Hence, building a robust and thus accurate and vigorous classifier for intelligent phishing detection in online transactions using website properties to analyse and detect phishing. This study is significant because the proposed system will enable online users to have confidence in performing their web activities with peace of mind. The remainder of the paper is organised as follows: Section 2 presents a review of the literature and related works. Section 3 describes the proposed intelligent system with the hybrid feature. Section 4 explains the extraction and analysis of website features. Section 5 covers the experimental procedure. Section 6 discusses the training and testing results. Section 7 concludes the paper by explaining this study's contribution to knowledge and outlining future work.

2. Literature review and related work

2.1. Literature review

Phishing is a type of online fraud in which a scammer uses a website or email to dishonestly obtain confidential information such as credit card details and Internet banking passwords (Martin, Anutthamaa, Sathyavathy, Francois, & Venkatesan, 2011). Phishing websites constitute a severe problem due to the enormous effect on the financial and online retail sectors. Due to the recent advances in technology, various procedures can now be used in phishing attacks that enable attackers to masquerade as a legitimate entity. Hence, they trick users into entering their credentials, such as passwords, username and credit card details into a fake web page for the attackers' malicious use (Babu, Nirmala and Kumar, 2010). Therefore, it is vital to prevent such attacks and defend against website phishing attacks (Aburrous et al., 2008). Khadir and Sony (2015) reviewed different kinds of phishing detection techniques mainly focusing on linguistic techniques and machine learning technology in a study that covered works up to 2015. Their review highlighted that various anti-phishing toolbars had been developed to protect Internet users. One such example is the eBay toolbar that helps users to monitor the web pages the users visit and provides warnings in the form of a coloured tab on the toolbar (eBay, 2016). Another is SpoofGuard, a plug-in developed for Internet Explorer, which examines the web pages users visit and warns them as to whether a particular page is likely to be a spoofed site (Neil Chou, 2004, Barraclough et al., 2013).

The use of non-executable files such as Microsoft Office and Adobe PDF documents attached to an email has been a component of many recent phishing attacks (Liao, 2008). Due to the failure of the filtering process of most email servers, anti-spam software and Internet mail clients, this type of attack has grown in popularity. Most email servers filter out any executable file attached to an email because of the risk they pose, but non-executable data can flow through and are considered safe by most users. Regrettably, a non-executable file constitutes a vulnerability that when exploited might allow a phisher to perform malicious actions on a victim's computer (Cohen, Nissim, Rokach & Elovici, 2016).

2.2. Content-based approaches

Aburrous et al. (2010a) propose a model based on fuzzy logic combined with a data-mining algorithm to characterise the e-banking phishing website. Their model indicates the worse e-banking phishing site rate of 83.7% e-banking phishing website. However, the feature set that needs to be more comprehensive because it is based on text only to detect the phishing of e-banking websites. As the number of features is based on text only, the process needs to include more web page features such as images and frames for identifying e-banking phishing sites to improve accuracy.

Likewise, Aburrous et al. (2010b) implement an intelligent and efficient model based on an association classification data-mining algorithm. This algorithm is used to identify rules and factors, to classify the phishing website and the relationship that correlates the elements and standards together. They implement six different techniques to extract text features from 120 sites to classify their legitimacy. The performance of each method is compared based on speed and accuracy. They demonstrate the feasibility of using each classification method in real applications involving large datasets and achieved better performance as compared to other traditional classification algorithms with an error rate of 12.62%. Nevertheless, they are unable to use different pruning techniques that remove the rules with the incorrect classification, which reduces the accuracy rate of the proposed method.

Barraclough et al. (2013) propose a neuro-fuzzy system with fuzzy rules to differentiate between suspicious, phishing and legitimate websites in real time. Their result shows higher accuracy with 2-fold cross-validation of 98.5% for real positives and 1.5% for false positives. This approach demonstrates the effectiveness of the neuro-fuzzy system when using five inputs for detecting phishing websites with higher accuracy in real time. Their comparison

mechanism is better compared to another study because of its improved efficiency. However, the proposed technique only uses text-based features and could be improved if more features were added and the parameters were optimised for more precision. These observations will form the basis of our proposed intelligent phishing detecting scheme.

Furthermore, Barraclough et al. (2015) develop an online toolbar that continuously runs in the background of the Internet Explorer web browser checking all the websites the user requests against a set of data in real time. Their approach uses a neuro-fuzzy scheme with six inputs: good site rules, user behaviour profile, PhishTank, user-specific sites, pop-up windows, and user credential profile for detecting phishing web pages. Their result reflects improvement in phishing detection in real time. However, they concentrate only on text-based features. Therefore, their work could be expanded further by analysing frames and image features to achieve better accuracy. The toolbar is developed using 300 full features based on six sets of inputs. This data is fed into the feature extractor algorithm based on neuro-fuzzy. The toolbar compares web page requests against elements and copies the website if a suspicious site is detected, and a text directive is generated in a red colour to alert the user. In testing, the solution achieves a 96% true positive rate and 4% false positive rate. The result is compared with the results of other work such as SpoofGaurd, Netcraft, EarthLink, Google, Cloudmark, Internet Explorer 8, TrustWatch and McAfee. The toolbar outperforms these regarding accuracy in real time. The main contribution of their study is the introduction of a novel voice-generated user warning interface algorithm for toolbar detection; this method will be integrated into our system implementation.

Barraclough and Sexton (2015) utilise six input data sources based on neuro-fuzzy and extract 300 features to train and test a fuzzy inference system with 2-fold cross-validation. The rationale for using a feature-based approach is to find useful features to generate fuzzy rules and classify legitimate, phishing and suspicious websites. The overall accuracy of their method is improved, and it outperforms other machine learning algorithms (Barraclough and Sexton, 2015). However, they use only text-based features, so this method could be improved by including frame and image features for more robust analysis and accuracy.

2.3. Visual similarity approaches

Wenyin et al. (2005) develop a solution based on the visual similarity between two web pages. They measure four metrics in their plan; web page layout, frame, block level, and overall style similarity. The block level similarity measures the visual similarity of two web pages and calculates the similarity in the feature set using the weighted sum of the individual similarity as the total sum of the two blocks. Hence, they focus on the visual style of the web page, including features such as fonts, background colour, line spacing and text alignment. They collect these features and calculate the normalised correlation coefficient of the two web pages' histograms. Even with this promising solution, more features and thorough testing are required for a more comprehensive result.

In a similar vein, Fu et al. (2006), Chen et al. (2010) and Zhang et al. (2011) use an earth mover's distance (EMD) algorithm to train a threshold vector for classifying a web page as legitimate or phishing. They measure the web page visual similarity by using EMD in their calculation. The EMD is used to calculate the signature distances of the images of the Internet pages. They demonstrate the advantage of their method by comparing it with region matching and HTML/DOM-based visual similarity methods for phishing detection. However, even with the use of EMD, the solution cannot detect a web page that does not have visual similarity. Therefore, there is a need to include more features to improve detection performance.

A scheme is also proposed by Dunlop et al. (2010), called GoldPhish, which provides zero-day protection against phishing attacks with high precision. However, the scrutiny time in this scheme is quite high. The approach uses a browser plug-in to detect and report phishing sites that use optical character recognition (OCR) to read the text from an image of the web page, and that grabs the top-ranked domains from a search engine. The solution captures the current web page in which the user is surfing as an image and use OCR techniques to convert the picture into readable text. Hence, the texts are collect and input into a search engine to retrieve page ranking results. Their application compares the top-level and second-level domain of the web page the user is visiting with the first four in the Google search engine results. When a match found, the application can verify the site and notify the user via the GoldPhish toolbar about the legitimacy of the website. In testing, the solution achieves 98% true positive and 2% false positive rates on over 100 sites. However, this method is unstable such that the image could be manipulated with little variation causing the process to fail regarding accuracy. Their solution is better than the previous image comparison method that relies on a database. Nevertheless, more comprehensive testing needs to be carried out because 100 sites are little compared to the number of phishing sites that appear every day.

Kumar and Kumar (2015) develop an anti-phishing solution based on a visual cryptography approach. In their method, the user generates two shares of images using a (2, 2) visual cryptography scheme. The first time that the user registers on a website, the user stores the first share of the image, and the other part is uploaded to the site. During each login attempt, the user must verify the legitimacy of the location by comparing the image of both shares (Kumar & Kumar, 2015). However, their test result is not robust due to the low number of websites used in the test experiment, so there is a need to undertake a comprehensive analysis to improve precision.

Likewise, Chou et al. (2004) developed a plug-in called SpoofGuard. SpoofGuard is an Internet Explorer browser plug-in. The plug-in accesses the Internet Explorer history files and uses three additional types of data stored in the user's profile directory. One of the records is a read-only file such as Hotmail and Yahoo Mail. The other two documents are the hash password and hash image history. This data is used by the plug-in to verify the authenticity of the web page. However, the toolbar plug-in solution specifically developed to reduce phishing attacks. The plug-in is symptom-based in that it looks for a phishing indicator, such as masked links, on the website and the similar domain name that the user visits. Their model with the password checks is critical to the strength of their solution. The image checker hash algorithm needs improvement to be able to detect small modifications to images. Alerts are generated based on the number of signs identified.

Similarly, Fatt et al. (2014) propose an approach that is based on the website favicon to check the identity of a site with the use of the Google search-by-image API search engine solution and use its result to evaluate the genuineness of a website. They use Google search-by-image, which is a content-based image retrieval technique for querying and get a list of information related to an image. The favicon is a website shortcut icon displayed on a web browser address bar, and it is extracted using four different methods (Fatt et al., 2014). The plans include a path to the favicon, reference link to the image, alternative attribute link to the picture and an Apple-touch-icon employed by a web server to display the favicon on a browser. These paths are pasted into the Google search-by-image API, and it returns a list of websites related to the favicon. Hence, they use four different heuristics in their method to classify a suspicious site based on the search results. The result

verifies the legitimacy of the site according to the page ranking. The first four on the page ranking list are considered to be original sites rather than phishing websites because they have a higher page ranking. They use 1,000 web pages to verify the effectiveness of their approach. The result shows that the method achieves 97.2% true positives and 2.8% false positives. The process has excellent precision but depends solely on the favicon for its analysis. The favicon of a legitimate website could be copied and used on a fake Internet site, which could result in high correct positive prediction. The solution is a bit reliable but may reduce accuracy, because if a website does not have a favicon, their method will regard it as a phishing site. However, there are some limitations to the technique, which could be detrimental as phishing activities evolve, so this method could achieve a better result if combined with other phishing detection features, such as text and frame structure.

2.4. Heuristic based approaches

Xiang et al. (2011) propose a web browser toolbar that uses a TF-IDF[3] algorithm with another website heuristic feature for detecting phishing. They evaluate their content-based approach, called CANTINA, which makes use of TD-IDF for identifying phishing sites. Their results show a 97% correct positive rate and 3% false positive rate. However, the approach was unable to distinguish between spam and phishing URLs.

Equally, Zhang and Yuan (2013) implement phishing detection through a feed-forward neural network by incorporating some essential features of the email structure and external link. They argue that the neural network is excellent in detecting phishing email and only misclassifies a small amount of non-phishing emails (Zhang & Yuan, 2013). However, their results fail to achieve a high accuracy rate. In their work, they write a script to extract text features from the body of emails and generate a feature vector set together with the best value into one text file. The best benefit from these properties is normalised before applying a machine learning algorithm. A total of 4,202 legitimate emails and 4,560 phishing emails are used in their study. Their implementation of a multilayer feed-forward neural network is in Java, which provides a framework to conveniently construct neural networks and perform training and testing of the dataset. They achieve 95% true positive classification, which implies that the neural network is excellent at detecting phishing emails with only a small amount of misclassification of non-phishing emails. Their implementation is improved method that gives a clear result with an explanation of the classification while comparing with other machine learning algorithms.

Lastly, Shekokar et al., (2015) also propose a solution for the detection and prevention of phishing that involves web page similarity and URL-based discovery. They use the LinkGuard[4] algorithm to analyse the extracted URL from which the website is directed and the virtual URL that is seen by the user. If phishing is not detected in the URL-based detection approach, the algorithm then proceeds to visual similarity-based detection (Shekokar et al., 2015). Unfortunately, their test result is not robust due to the low number of websites used in their test experiment, so there is a need to conduct a comprehensive analysis to improve precision.

2.5. Why use an adaptive neuro-fuzzy inference system?

The ANFIS has been used for decades in the engineering sciences to embed expert input into a computer model for a broad range of applications. It offers a capable alternative for determining operational risk. The integrated of the neural network and fuzzy inference systems formulated into the synchronised neuro-fuzzy algorithm. This algorithm loads its essential component in rule-based and performs fuzzy reasoning to deduce overall output value (Abraham, 2005). The ANFIS is perhaps the first integrated neuro-fuzzy model, and its typical architecture illustrated in Fig. 3. There are two types of fuzzy inference system models: the Mamdani and the Sugeno models (Karaboga & Kaya, 2016). These fuzzy inference systems have two inputs and one output. Mamdani's neuro-fuzzy system uses a supervised learning technique, back-propagation learning, to acquire the parameters of the membership functions (Abraham, 2005).

To the best of our knowledge, none of the published literature has used text, frame and image feature together to detect phishing websites automatically in real time. Therefore, there is room for improvement of several components, such as text feature, frame and image feature extraction. The combination of these three sets of elements should enhance detection and improve security for users when carrying out transactions on the Internet. Hence, our study intends to develop a robust algorithm to address the alarming rate of a phishing attack and provide a comprehensive solution.

The Table 1 below present list of phishing plugin, the indicative of techniques they with their effectiveness and service. Each of the plugins was developed for a specific browser, and not all are built for a cross-platform for the application. That has shown some weakness in the build has end user has to choose a browser that he is not used to when accessing contentment on the Internet.

Table 2, also shows the type of features each browser in Table 1 uses for their phishing detection. Column 1, which represent the list of phishing plugin in Table 2 below. The rest of the column illustrates the type of features each plugin is using in their detection model. As indicated in the table, the majority of the plugin uses text and heuristic approach in their scheme. The heuristic-based anti-phishing technique uses website features such as text and frame content for its phishing detection analysis with the aim of creating a robust classification model (Lee & Park, 2016). Other uses blacklist/whitelist approach that is like the use of signatures in antivirus that maintains a blacklist of the site that contains malicious content. Blacklisting is reactive and can be evaded by rapid recycling blocked phishing web page. However, in our solution, all the features and techniques listed in Table 1 will be exploring to develop the phishing detection and protection scheme. As this has not been used together as a single solution in any of the previous approaches and this is the strength of plugin development.

3. The proposed intelligent phishing detection and protection scheme

Here, we introduce the proposed intelligent phishing detection and protection system (IPDPS). We also look at the different issues that arise in detecting phishing websites. This section covers phishing detection implementation using sets of the dataset, the essential characteristics and features of phishing website extraction techniques.

3.1. The main component of intelligent phishing detection and protection scheme

Developing with the anti-phishing methods, phishers use various phishing methods and more complex and hard-to-detect approaches. The most straightforward way for a phisher to swindle people is to make the phishing web page similar to their target. However, many distinctive and features can distinguish the original legitimate website from the clone phishing website like the spelling error, image alteration, long URL address and abnormal

[3] TD–IDF, is short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining.

[4] LinkGuard algorithm [Online] Available at: < http://www.ijafrc.org/Volume3/issue34/6.pdf > [Accessed 20 October 2016]

Table 1
List of anti-phishing plugin and their effectiveness.

Plugin for Phishing	Features & Techniques	Browser	Effectiveness %	Service Type
GoldPhish	Heuristics & Features-based	IE	98	Free
Cloudmark	Heuristics	IE	94	Free
Microsoft SmartScreen	blacklist and whitelist	IE	95.9	Free
Netcraft (Customise)	blacklist and whitelist	Chrome; Firefox	90	Free
SpoofGaurd	Heuristics & Features-based	IE	91	Free
Phishdentity	Google search-by-image API	IE	97.2	Research
PhisTackle	Heuristics & Features-based	IE	91.3	Research
PhishGuard	Heuristics	Firefox	94	Research
PhishIdentifier	Heuristics	Firefox	92	Research
PhishTester	Heuristics & Features-based	IE	97.1	Research
CANTINA+	Heuristics & Features-based	IE	98.06	Research
PhishAri	Features-based	Chrome	92.52	Research
PhishShield	Heuristics & Features-based	Chrome	96.57	Research
PhishNet	blacklist	Chrome	95.0	Research
PhishDef	Heuristics & Features-based	Chrome	97	Research
Google safe browsing	blacklist	Chrome; Firefox	93.3	Free
PhishZoo	Heuristics	Chrome	96.10	Research
Seclayer	Heuristics & Features-based	IE; Chrome	91	Free
IPDPS	Heuristics, Features-based & image	IE; Chrome; Firefox	98.55	Research

Table 2
Techniques and feature for phishing detection.

Phishing Plugin	Techniques/Features					
	AI	Frame	Heuristic	Image	Text	Whitelist & Blacklist
GoldPhish			√	√	√	
Cloudmark			√		√	
Microsoft SmartScreen					√	√
Netcraft (Customise)						√
SpoofGaurd			√		√	
Phishdentity			√	√	√	
PhisTackle				√	√	
PhishGuard			√		√	
PhishIdentifier			√		√	
PhishTester			√		√	
CANTINA+			√		√	
PhishAri					√	
PhishShield			√		√	
PhishNet						√
PhishDef			√			
Google safe browsing						√
PhishZoo			√		√	
Seclayer			√			
IPDPS	√	√	√	√	√	√

DNS records. The full list is revealed in Table 3 which is used later in our analysis and classification study. If an attacker clones a legitimate website as a whole or designed to look similar as they usually do in most attacks in recent times, our approach is that similar looking phishing web page content is not left for the users to check for the indicator or the authenticity attentively, but can detect by automated methods. Our approach is based on website phishing detection using the features of the site, content and their appearance. These properties are stored in a local database (Excel table) as a knowledge model and first compared with the newly loaded site at the time of loading against the dangerous web page offline. After the comparison was unable to detect the similarity, then the critical approach to compare the legitimate and fake using the features of the website with machine learning for an intelligent decision. The critical contribution of our approach includes:

The method that depends solely on websites' content to detect similar phishing site. It can discover newly phishing website that is not yet blacklisted and targeted against unsuspected users. We explore dissolute, online detection using both the HTML and URL content in the properties of a website. Our method can detect 98.55% of current phishing sites, with 1.45% false positives. We as well study vision techniques to identify phishing website more vigorously. This technique requires both scene analysis and images matching. Our solution explores the matching problem, which is sufficient to identify new phishing sites. Using the Scale-invariant feature transform (SIFT) image matching algorithm, the approach can identify 97.2% of phishing sites with a false positive rate of 2.8%. This method can be used offline by an application for more quickly discover phishing websites.

Hence, various experiments are conducted to gather and analyse phishing factors with relevant rules. The main advantage offered by the ANFIS model is the use of linguistic variables to identify phishing features in use for the building of our intelligent phishing website detection model. The overall outcome of this study is a practical plug-in phishing toolbar implementation with appropriate testing and validation.

The model that most frequently used with the ANFIS is the Sugeno fuzzy model. The Sugeno model is a zero-order model. This model will be employed in this research experiment because it is the only one that can be implemented with the ANFIS, it has differentiable functions that can learn the fuzzy primary system from data and it can be easily understood (Abraham, 2005). The detailed features of each layer of the zero-order Sugeno fuzzy inference system with inputs x and y and two rules are as follows:

IF x_1 *is* A_1
AND x_2 *is* A_2.

Table 3
Hybrid features.

Text Features Approach	
	Search index
Page ranking	This feature was used to check the importance of the web page by counting the number of quality links to a page to determine the relevance of the site on the Internet.
Google index	This feature was used to compare if the URL of the website included in the Google index matched the one submitted to google index.
Website traffic	This feature is used to measure the amount of data sent and received by a visitor to a website.
Statistical-report	This feature is also used for the usage of the website such as the number of queries and the website availabilities. However, a new website may fail this check; some other features are used to ascertain the legitimacy of a website.
	Security & encryption
Long URL	This feature was used to check the length of URL to determine if the original website has the correspondent URL.
Using the IP address	This feature was used to check the URL if it contains IP address as most phishers use this to deceive the unsuspected user.
Abnormal URL	This feature will check the URL against abnormality in the resources locator against the information store in WHOIS database for the legitimate website.
Abnormal request	This feature checks if there is a request from an external object within the web page such as image or video loaded from another domain.
Abnormal Anchor	This feature checks if their anchor element is like a tag ⟨a⟩ from an external link. This feature is treated as the request URL.
	Web address bar
Adding prefix or suffix	This feature is used to check if dash symbol that is rarely used in a valid URLs. Phishers tend to add suffix or prefix to separate by (-) to the domain name to made users feel that they are dealing with the legitimate web page. These are checked in the URL with our approach.
URL is having "@" symbol	This feature is used to check for the @ symbol in the URL as it leads the browser to ignore everything preceding the @ symbol
Using URL shortening services	This features check for considerably smaller URL length and still leads to the acquired web page. These are achieved by using https redirect on a domain name that is short.
Some links are pointing to a page	This feature checks the number of links that are pointing to the web page.
Using non-standard port	This feature is useful as it checks for validating if a service such as https is up or down. If all ports are open, phishers can run almost any service they want. As a result, user information is threatened.
	Domain identity
Age of the domain	This feature is used to extract the information from WHOIS database and compare with information of a phishing site. Most phishing websites live for a short period.
DNS record	This feature was used to check the identity of the domain in WHOIS database for the records. However, If the DNS record is not found or empty, the website is then classified as a phishing web page.
Domain registration length	This use of this feature is to check how the site is registered. Since phishing websites live for a short period, we believe that trustworthy domains are usually paid for several years in advance.
	Source code Javascript
Redirect using "//"	This feature was used to check the existence of // within the URL path, which means that the user will be redirected to another website.
Submitting information to email	This feature was used to check if a website redirected user's information to a personal email, instead of a server to process.
https	This feature is used to check the existence of secure communication and if the issuer is trusted and how long, the certificate is issued.
	Frame Features Approach
Iframe redirection	This feature is used to check the HTML tag used to display additional web pages in the current website. A phisher will take advantage of it by making the tag invisible without a frame border.
Disabling right click	This feature is used to check if the right-click function is disabled using the JavaScript so that users cannot save or view the web page's source code.
Using pop-up window	This feature is used to check if users were asked to submit their personal information through a pop-up window, which is unusual to find in a legitimate website.
Server form handler (SHF)	This feature is used to check if the domain name in server form handler is different from the domain name of the web page
Website forwarding	This feature is used to check how many times a website has a redirect, a legitimate site does one time, while phishing site repeats this more than four times.
The link in Script & Meta	This feature is used to check that the tag on the website is linked to the same domain of the web page.
Layout similarity	This feature is used to check the percentage of the layout similarity of the web page.
Style similarity	This feature is used to check the percentage of the style similarity of the web page.
	Image Features Approach
Favicon	This feature is used to check the icon associated with a particular web page and check if the icon is loaded from a domain other than that is shown in the address bar.
Image size	This feature is used to check the size of the images on the website
Alternative text	This feature is used to check with some level percentage if alternative text is used on the website
Mouse over	This feature is used to check if JavaScript is used to show a fake URL in the status bar to users.
Login form	This feature is used to check if there is an obstructive login form on the website

D x_m *is* A_m

THEN $y = f(x_1, x_2, \ldots x_m)$ where $x_1, x_2, \ldots x_m$ are input variables; $A_1, A_2, \ldots A_m$ are fuzzy sets, and y is either a constant or a linear function of the input variables. However, if y is constant, the zero-order Sugeno fuzzy model which the resulting of rule-based is specified could be obtained by singleton (Barraclough et al., 2013).

The ANFIS is a hybrid learning algorithm that can be used to tune the parameters of a Sugeno-type fuzzy inference system. The algorithm uses a combination of the least-squares and back-propagation gradient descent methods to model a training dataset. Hybrid features are used because they can represent phishing attack techniques and strategies. These features are used as training and testing input data for the neuro-fuzzy inference system so that it can generate fuzzy *IF…. THEN* rules to differentiate between legitimate, suspicious and phishing websites.

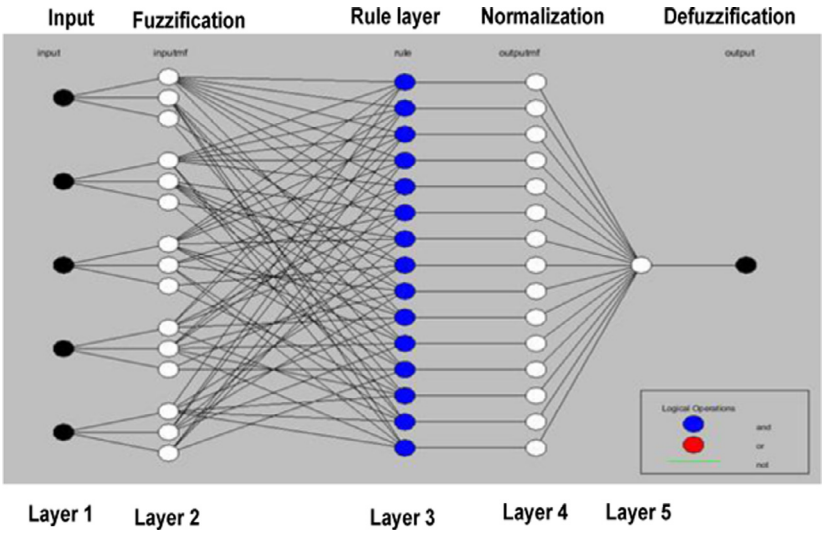

Fig. 1. Structure of intelligent phishing detection fuzzy inference system (Barraclough et al., 2013).

The structure of the intelligent fuzzy inference system has five function layers used in making the decision are as follows: a) Input layer

Each node in this layer is assigned parameter, which includes three membership functions. The neurones in this layer quickly transmit visible crisp indications straight to the next tier (Kim & Kasabov, 1999). The Eq. (1) below shows the measure in which this is done where a_i,b_i,c_i $(i = 1, 2, \ldots. nth)$ is the parameter set, $\mu_A(x)$ is the membership function of fuzzy seta A_i and X is the input. As the value of the parameter change, the shape of the bell-shaped function varies (Fig. 1) and is referred to as premise parameters (Cruz & Mestrado, 2009)

$$\mu_A(x) = \frac{1}{1 + |\frac{x - c_i}{a_i}|^{2b_i}} \tag{1}$$

b) Fuzzification layer.

A node in this layer acts as a membership function to represent the terms of the respective linguistic label such as phishing, suspicious and legitimate it assigns a value for each key phishing feature indicator (Dariane & Azimi, 2016). The valid range of the inputs (X, Y) considered that divided into the fuzzy set. The output value in the input layer is fed into this layer, and Gaussian membership functions are used with two parameters, variance and mean. Also μ_{A_i} *and* μ_{B_i} is the grade of the fuzzy sets. The output function of this node is the product to which the input belongs and the given membership function.

$$O_i^2 = w_i = \mu_{A_i}(x) \cdot \ \mu_{B_i}(y), \ i = 1, 2 \tag{2}$$

where A and B are the input parameters used in the equation. Every node in this layer is a fixed node, whose output is the product of all the incoming signals. Each node's output represents the firing strength of a rule (Cruz & Mestrado, 2009). The output is used to determine the number of rules in next layer. A sample of rule-based functionality presented in Table 3. c) Rule generation layer

Every node in this layer is a fixed node labelled N. The i^{th} node calculates the ratio of the i^{th}rule firing strength to the sum of all rules' firing strengths. That done by a rule-based layer that consists of *IF…THEN* statements that are related to phishing site opportunities at different levels that get input w_i from the individual fuzzification i^{th}nodes and calculate the strength of the rule it represents (Karaboga & Kaya, 2016).

$$O_i^3 = \overline{w_i} = \frac{w_i}{w_1 + w_2}, \ i = 1, 2, \ w_i \ = \text{input} \tag{3}$$

The output of this layer is called normalised firing strengths. d) Normalisation layer

This layer is where normalisation occurs. All the neurones in this layer are connected to an individual normalisation neurone, as illustrated in Fig. 1. The output neurone from the rule-based layer is fed into this layer and normalise firing strength is resolved. The power of the normalised firing neurone is the percentage of the firing strength as instructed and the sum of the firing force of every rule (Barraclough et al., 2013).

$$O_i^4 = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i), \tag{4}$$

where $\overline{w_i}$ is the normalised firing strength from layer 3 and $\{p_i, q_i, r_i\}$ are the parameters settings, which are referred to as essential parameters (Cruz & Mestrado, 2009) e) Defuzzification layer

This layer is where defuzzification occurs. The neurone combines the sum of all the output neurones and produces the ANFIS output as shown in Fig. 1. The single node in this layer calculates the total output as the summation of the contribution from each rule. The input for the defuzzification process is the aggregate output of fuzzy set, and a result is a number. The output of the phishing web page and risk rate is defined in fuzzy sets as legitimate, suspicious, phishing. The output set is then defuzzified to arrive at a scalar value.

$$O_1^5 = Overall \ \ output = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{5}$$

The proposed approach uses the ANFIS with hybrid feature inputs to detect phishing websites for online transactions together with a knowledge model (Fig. 2) to maximise the accuracy, reduce the false positives and improve the operation time. The overall intelligent phishing detection block structure is presented in Fig. 2 below. The concept behind the above Fig. 2 involves integrating the extracted features in websites to predict phishing activities, using a web browser plug-in. The phishing classification steps, consisting of the image, frame and text features, the machine learning, and classification will use the ANFIS as it applied in the initial stage.

The output is determined by the classifier, in the phishing detection stage which predicts if the web page is suspicious, legitimate or phishing. The knowledge model and plug-in development will be developed at a later stage. The phishing detection block diagram in Fig. 2 is explained further by Fig. 3, which illustrates the process of acquiring the website features and feeding them into the fuzzy inference system for training and testing purposes;

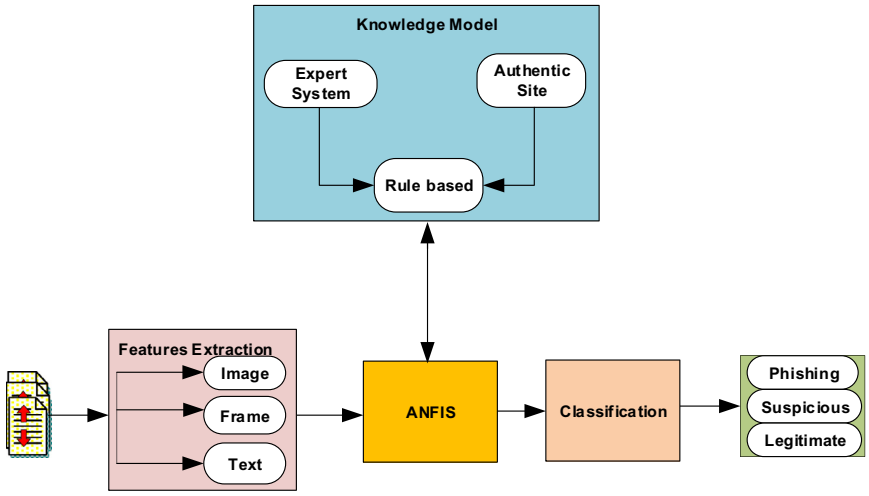

Fig. 2. Conceptual diagram of smart phishing detection system based on image text and frame fusion.

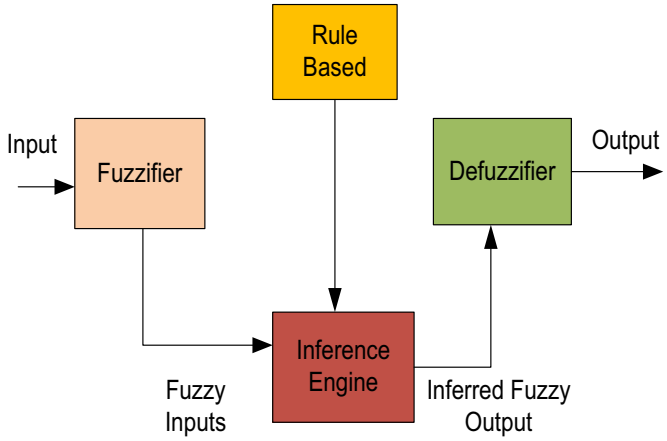

Fig. 3. Diagram of intelligent phishing fuzzy inference system structure.

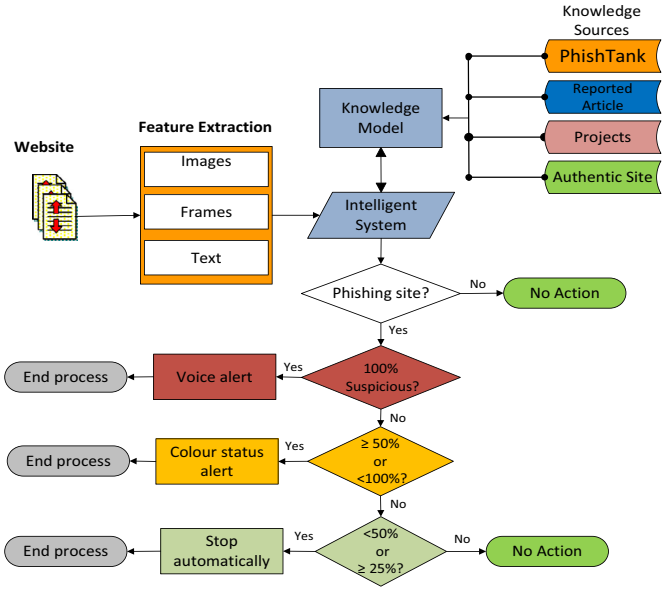

Fig. 4. Process diagram of intelligent phishing detection and protection scheme.

then the fuzzy *IF…THEN* rules are applied to distinguish the legitimate, suspicious and phishing websites accurately in real time at the output step.

Fig. 4 presents the conceptual system design and the classification of phishing website using text, image and frame features based on web page properties. A web browser plug-in will be developed using the classification algorithm, which would run in the background of the Internet browser and check all websites requests to distinguish between legitimate, suspicious and phishing sites.

Fig. 4 shows the system structure is composed of five components: website analysis and feature extraction, intelligent system, knowledge model, knowledge sources and output process

The plug-in uses the ANFIS *IF…THEN* rule and the features from three sets of web page properties that combine the significant elements for plug-in development in such a way to detect phishing and legitimate sites accurately in real time. If a phishing website is detected, a sound alarm is generated to alert the user. As well-established that subjective judgment of perceived urgency with reaction time (Suied et al., 2008). If the site is suspicious, a red colour status with a text-based risk explanation is made to inform the user. Because red is associated with the danger or failure in achievement contexts and evokes avoidance motivation (Elliot, Maier, Moller, Friedman, & Meinhardt, 2007). In a situation where the rule violated, a warning is generated to alert the user, that is a proportion of its warnings genuinely indicate the condition to be avoided (Getty et al., 1995). However, if the information is less severe, a green colour status is made, and the user can continue, but if it is highly likely that sensitive information was stolen. Then the process is automatically stopped, which prevents the user's personal information from being acquired by the fake website, as illustrated in Fig. 4.

4. Features selection and detection criteria analysis

In this study, reviewing different phishing investigations, research papers, conducting a separate phishing experimental case study give us more insight into the selection of the feature used for our phishing classification. Given this, we can extract 35 elements and factors which characterise the signature of any phishing website incident. These datasets divided into seven criteria that are distributed into three layers, depending on the attack type. The most popular feature selection methods in the literature are Chi-Square (x^2) and information gain. Chi-Square is the commonly used method and adopted in this study; evaluated features are by computing Chi-Square statistics on classes (Gaunt, 2016). The information gain technique is also used in feature selection, which decreases the size of features by calculating the value of each at-

tribute and ranking them. In other words, information gain selects elements through scores (Zeng et al., 2016). In this approach, a subset of initially chosen features is only used for testing and training the classifier (Abunadi et al., 2013).

Feature extraction usually converts the original feature space into a more compact space. However, the original features are retained and transformed into a new reduced space with only a few representative sets (Zareapoor & Seeja, 2015). This approach mainly uses principal component analysis (PCA) and latent semantic analysis (LSA). Principal component analysis reduces the dimension of the data by transforming the actual attribute space into a smaller one (Vidal et al., 2016). That could achieve by converting the real variables $Y = [y1, y2, \ldots, yn]$ (where n is $n^{\tan h}$ number of actual variables) into new variables, $T = [t1, t2, \ldots, tp]$ (where p is an n^{th} number of the new set of variables). The LSA technique is a novel-based method of text classification. The approach analyses the relationship between a concept and term contained in unstructured data, and it has the ability to correlate semantically related value that are latent in nature (Marcolin & Becker, 2016). These processes are used to convert the hybrid features into data that can be used to train and test our model.

4.1. How features are chosen

Selecting phishing features required cautious reflexion. The use of primary and secondary sources enables features extraction based on text, image and frame as follows: Specifically, on text-based elements consist of 22 extracted components, which are selected based on the effectiveness in detecting phishing and reduction in feature redundancy by exploring document and journal paper. Equally, frame-based features consist of 8 features extracted by studying several legitimate, suspicious and phishing websites (PhishTank, 2017). Likewise, image-based features have 5 features which were gathered by observing the image of both legal and phishing websites (Abunadi et al., 2013). That makes the overall total of 35 features also known as data used to test our model.

4.2. System detection criteria

The selection of phishing features requires careful deliberation. The use of primary and secondary sources for feature extraction based on images, frames and text are described below.

This section presents a reasonable phishing detection rate performed based on seven criteria: Search index, URL content, web address bar, image identity, domain identity, source code & javascript and page style & layout identity, as shown in Fig. 5. The Fig. 5 also shows that there are different number of components for each criterion. The search index has four elements, and URL content has six parts, while there are five elements for the web address bar and image identity, four elements for domain identity, three features for source code & javascript, and eight for page style & layout identity. Therefore, there are 35 critical components in total. The elements are selected as the best for the detection of phishing and improve the time of discovery too. The proposed intelligent phishing detection scheme has three layers, as illustrated in Fig. 5. The first layer contains only the text identity component with the search index criterion with a weight equal to 0.2, while the URL content criterion is assigned a weight equal to 0.3 for it essential, as a user follow this link to their vulnerable site. The web address bar, image identity, domain content, page style & layout and source code & JavaScript each have a weight equal to 0.1.

The seven criteria have been prioritised according to their importance using weights to rate them as concluded from a case study, website phishing experiments, Anti-phishing tools analysis, web survey, phishing quizzes, phishing expert feedback and detailed questionnaire. Various parameter values were used for the most efficient detection approach, but below parameter values provide the best result in our model.

Text crisp is represented as follows:

d_1 = URL Content

d_2 = Search index

d_3 = Security & encryption

d_4 = Domain identity

d_5 = Source code & JavaScript

The Frame crisp is represented as follows:

g_1 = Page style & Layout identity

The Image crisp is represented as follows:

h_1 = Image identity

The intelligent phishing detection rating Z_1 weight parameters is calculated as:

$$z_1 = (0.3 * d_1) + (0.2 * d_2) + (0.1 * d_3) + (0.1 * d_4) + (0.1 * d_5) + (0.1 * g_1) + (0.1 * h_1) \quad (6)$$

However, to use the right phishing features, it is also essential to take into consideration that phishing strategies and techniques change with time. The number of features that could be used for modelling can vary. There are some challenges attached to phishing websites post-classification. The most challenging concerns the phishing website material and date as a form of information, which has the net effect of increasing the false negative rate. The age of the dataset is the most substantial problem primarily regarding the phishing quantity. Some phishing websites are short-lived, sometimes lasting only 48 h. (Barraclough et al., 2013).

4.3. Dataset

Two existing independent datasets were used to test the application: The University of California Irvine, Rami et al., (2015a) and the University of Huddersfield, Rami et al. (2015b).

The datasets from University of California Irvine was collected in March 2015, has thirty attributes, for text-based features and frame-based features with a total of approximately 2,456 websites hits and total number 11,056 datasets. The elements are assigned weights, the value of 1 been legitimate, suspicious is assign 0, and phishing is assigned −1. Also, datasets from the University of Huddersfield (Rami et al., 2015b) is the same group of the researcher that donate to University of California. However, a different set of data that is an update in July 2015, it has the same thirty attribute comprises of approximately 2,700 datasets with the equal weights (−1, 0, 1) assign to the features.

Also, an offline selection of the image dataset was also collected using (SIFT) image matching to manage elements for image size, noise and illumination; these features were used to identify the object when attempting to locate the image in testing that contained many another object. These features are extracted and sore as dataset which used for training and testing. The phishing website data was collected from PhishTank and Anti-Phishing Working Group (APWG), between April to October 2017.

5. Experimental procedures

The goal of this study was to gather information about the approaches employed by hackers and to articulate theory about categorising and labelling all the different phishing website attack techniques. A dataset Rami et al. (2015) was selected containing features relating to the various methods that have been used for thoroughly investigating phishing attacks and how the use of these technologies has changed over time. In compiling the data, we found some exciting techniques that depend on the awareness that most users do not know how to check the security of sites that ask for sensitive data, which makes it difficult for users to see the

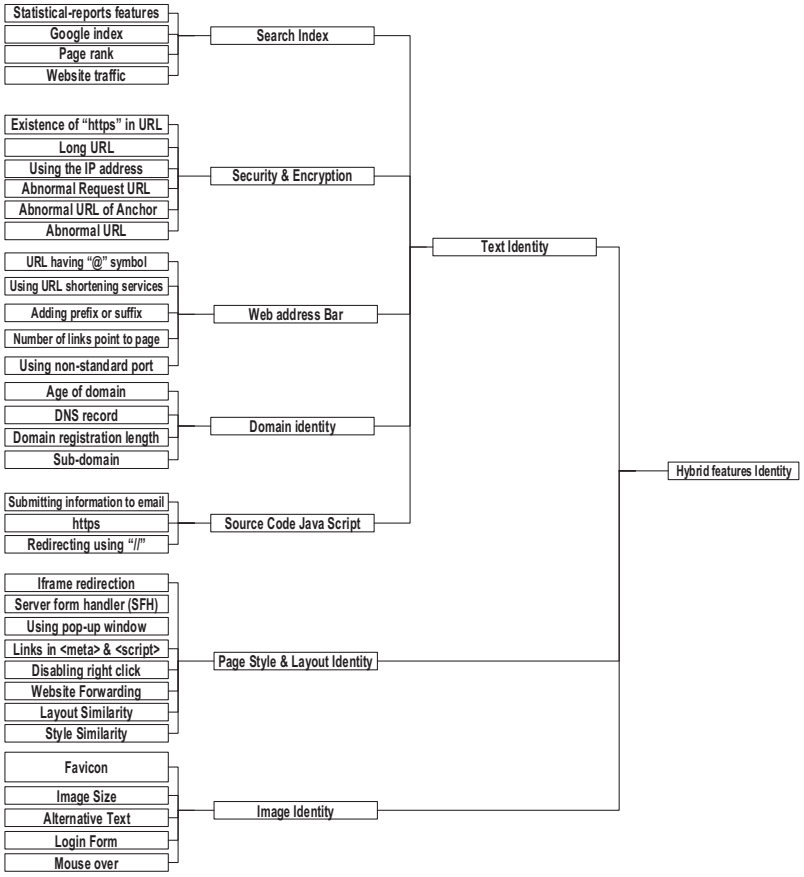

Fig. 5. The overall structure of ANFIS whole system for evaluating phishing websites.

difference between real security and impersonated security properties. Some findings also show that even sophisticated users often deceived by visual deception attacks (Aburrous et al., 2010).

A primary aim of the experiment was to improve the detection accuracy and robustness of the system while at the same time reducing scrutiny time. Various methods in the literature could be used for training fuzzy models, including 2-fold, 5-fold and 10-fold cross-validation. In this study, all the different folds of cross-validation (Barraclough et al., 2013) were employed to train and test the accuracy and robustness of the proposed model. The result showed that the 5-fold cross-validation method produced the best result and was due to its effectiveness on existing datasets. During training, the algorithm learns and extracts features from the data file, reads them and uses them to create fuzzy rules.

A total of 8,355 Rami et al. (2015a) and 2,500 Rami et al.. 2015b) with some 2,145-manual selection from PhishTank datasets were used in our classification implementation, and the data divided into five for training and a testing dataset. The overall dataset consisted of 4,898 phishing websites, 1,945 suspicious sites and 6,157 legitimate websites.

The practical aspect of this study lay in its utilisation of three different conventional classification algorithms (SVM, K-NN and ANFIS). The experimental result is presented in chart form in Chart. 1. The choice of classification method is based on the different strategies used to learn the rules from the datasets (Barraclough et al., 2013). The ANFIS uses linguistic terms and is represented by only one fuzzy set. The ANFIS does not provide the means to apply value that restrict the type of modification applied to membership functions, and it compared with a radial basis function network.

The SVM uses a clustering algorithm that provides a supervised learning model with an associated learning algorithm that analyses the data classification and performs regression analysis. The ANFIS approach is an improvement on the SVM, and it can efficiently deliver nonlinear classification using a kernel trick, implicitly mapping inputs into a high-dimensional feature space (Huang et al., 2012). K-NN is an instance-based algorithm that functions only by storing all available cases and classify new instances based on similarity measures, and all computation is deferred until classification is derived. The K-NN algorithm is among the simplest of all machine algorithms.

The proposed intelligent phishing detection and protection scheme was implemented in MATLAB Version: 9.1 using the ANFIS designer toolbox and classification learner toolbox for the experiment. The parameters in the neuro-fuzzy design toolbox were set to epoch 10, the error rate was set to 0, the optimisation method was established to hybrid (Fig. 6), and 5-fold cross-validation was used due to its effectiveness on existing datasets.

The input parameter was also assigned and included three membership functions. The generalised bell-shaped membership function was chosen for various ranges on the X-axis and Y-axis (Fig. 1). Improving the model's effectiveness and overcome the problems of operational time and high false positives, a hybrid method for parameter optimisation was applied with ten epochs and zero tolerance error set as illustrated in Fig. 6. The testing data and training data was present in the input layer. The neuro-fuzzy scheme uses the developed rules to differentiate between legitimate sites, phishing sites and suspicious sites.

According to our results, the classification approach is promising. The training and classification proved that it is possible

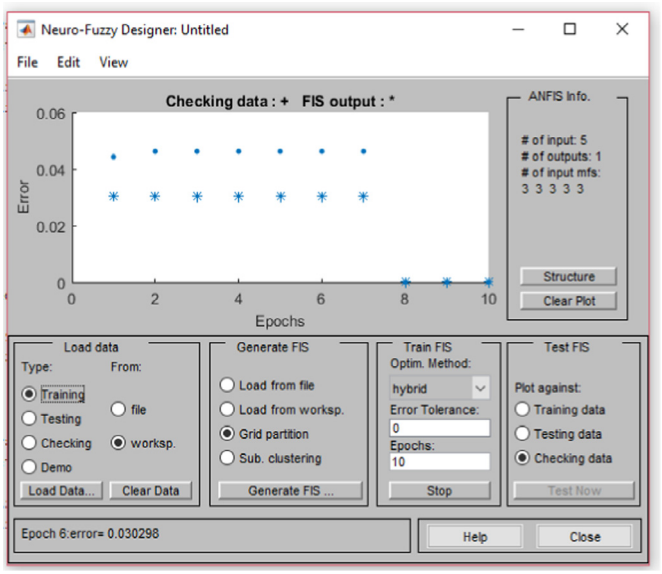

Fig. 6. Parameter settings for neuro-fuzzy designer training.

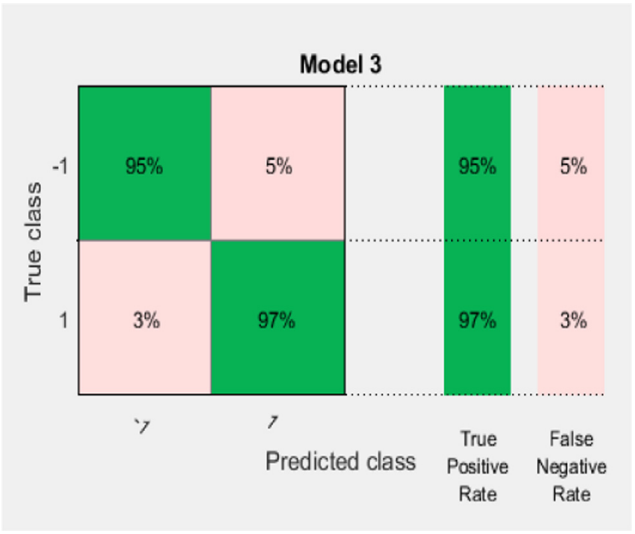

Fig. 7. Confusion matrix for phishing dataset.

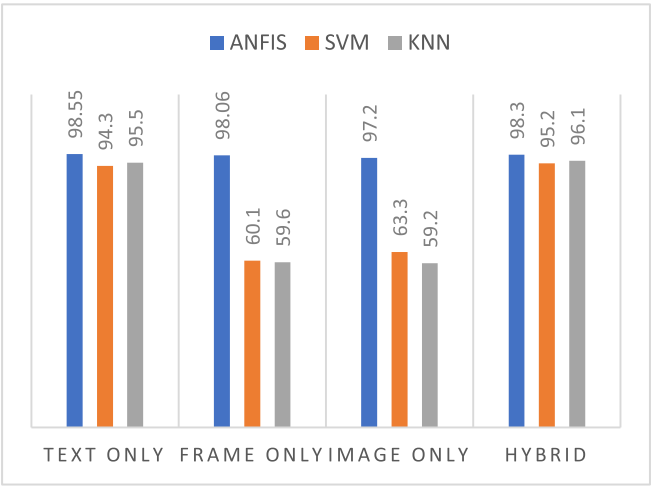

Chart 1. Experimental result for ANFIS, SVM and KNN classification.

to improve the categorisation process. The knowledge model in Fig. 2 will be employed and interfaced with the ANFIS model for active phishing detection. The next section presents the training and testing results.

6. Training and testing results with discussion

Training and testing were performed with the Rami et al. 2015a,b) dataset. The five-component input parameter (Table 1) also includes three membership functions. The process goes through inference systems and neural network, with the rules developed (Table 3) contain $(3^5) = 243$ entries to decide through the output layer and this repeated in a way that the dataset used once and the error rate recorded.

The training was done using supervised learning, with five input datasets split into training and endorsement data. The training datasets were presented through the input layer in an arbitrary order. The process goes through the fuzzy inference system and the neural network to decide which is achieved by reasoning about the rules that are provided in the rule base until the output layer reached.

Following training, the testing step followed the same procedure as the training step. The process is repeated for each input dataset such that the dataset is used once. The average error rate was calculated by summing up the error rate then dividing it by the number of data items, and the result was used to measure performance. The experimental results were compared with those of other classification algorithms, as shown in Chart 1.

Recall, accuracy, precision and *F*-measure can be calculated using a confusion matrix (Fig. 7). True positive (TP) denotes the number of phishing websites correctly classified as phishing, false positive (FP) signifies the number of phishing sites classified wrongly, true negative (TN) represents the number of legitimate websites ranked incorrectly, and false negative (FN) denotes the number of valid websites classified correctly. The matrix calculated as follows:

$$\text{Precision} = \frac{TP}{(TP + FP)}$$

$$Recall = \frac{TP}{(TP + FN)}$$

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

$$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

The features used in this experiment consisted of 35 predictors features and one response as it was mentioned before; these features represent the 22 text features, 8 frame properties and 5 image features. Table 4 through Table 9 present the result for text features, frame features; image features and hybrid features, respectively.

In the tables, Column 2 shows the accuracy achieved by using the data in the confusion matrix and using the calculation above to evaluate the effect. Column 3 shows the recall result, which is the proportion of real active cases that were predicted correctly as positive. These were measured by dividing TP by the sum of TP and FN then transforming that value into a percentage rounded to two decimal places. Column 4 displays the precision, which denotes the proportion of existent active cases that were predicted correctly as actuality. Precision is actually what machine learning, data mining and information retrieval focus on, but it ignored in receiver operation characteristics (ROC) analysis (Powers, 2011). Column 5 presents the *F*-measure result, which effectively references the TP

Table 4
A sample of rule-based intelligent phishing detection.

Rule#	(Component 1) Using IP address	(Component 2) Long URL	(Component 3) Short service	(Component 4) Using the at sign	(Component 5) Using double splash	Text features phishing risk
1	Low	Low	Low	Low	Low	Legitimate
2	Moderate	Moderate	Moderate	Moderate	Moderate	Suspicious
3	High	High	High	High	High	Phishing
4	Low	Low	High	High	Low	Suspicious
5	High	Low	Low	High	High	Phishing
6	Low	Low	Moderate	Moderate	Moderate	Suspicious
7	Low	Moderate	High	Moderate	High	Phishing
8	Moderate	Low	Low	Moderate	Moderate	Suspicious
9	High	Moderate	High	Low	Moderate	Phishing
10	Low	Low	Low	Moderate	Low	Legitimate
11	Low	Low	Moderate	Low	Moderate	Legitimate
12	Low	High	Low	High	Low	Suspicious
13	Moderate	Low	High	Moderate	High	Phishing
14	Low	Low	Low	Moderate	High	Legitimate
15	Low	High	High	Low	High	Phishing

Table 5
Classification of text features.

Algorithm	Accuracy %	Recall %	Precision %	F-measure %
ANFIS	98.55	98.51	98.58	98.54
KNN	95.50	95.45	95.54	95.49
SVM Quadratic	94.30	94.29	94.31	94.29

Table 6
Classification of frame features.

Algorithm	Accuracy %	Recall %	Precision %	F-measure %
ANFIS	98.06	98.02	98.08	98.02
KNN	59.59	59.20	59.60	59.39
SVM Quadratic	59.99	59.90	60.10	59.99

Table 7
Classification of image features.

Algorithm	Accuracy %	Recall %	Precision %	F-measure %
ANFIS	97.20	97.18	97.22	97.18
KNN	59.20	59.19	59.21	59.20
SVM Quadratic	63.30	63.29	63.32	63.30

Table 8
Classification of hybrid features.

Algorithm	Accuracy %	Recall %	Precision %	F-measure %
ANFIS	98.30	98.26	98.31	98.28
KNN	96.10	96.05	96.14	96.09
SVM Quadratic	95.20	95.18	95.23	95.20

to the arithmetic mean of the predicted positives and the real positives in proportion to a specific agreement in the actual class and the set-Dice coefficient (Hripcsak & Rothschild, 2005). The overall training accuracy was 98.55% for text features. These processes were repeated for the frame features with an average training accuracy of 98.06%, while image features produced an average efficiency of 97.2% and hybrid features had an overall training error of 98.3%.

The above Table 5 and 6 reflect the fact that the KNN and SVM have a low accuracy result and detection time of 52.6 seconds due to the dataset used and the features selection compare to the ANFIS algorithm. The two algorithms were chosen based on the usage in some top journal for classification.

The 35-features set was divided into seven separate subsets (Table 8) with five elements in each subgroup. Each loaded is the subgroup into the input layer in a random order to commence the training process, then back-propagation corrected the errors, and the error rate recorded. The overall average training accuracy result for hybrid features was recorded as 98.3% (Table 7) in the experiment using 5-fold cross-validation with an average time of 26.72 seconds. The best relative performance result achieved in training and testing compared to the rest of the tested unit is 98.55% for text-based cross-validation (Table 9). In this paper, 5-fold cross-validation method was employed to train and test the robustness and the accuracy of the proposed model (Table 10). The 5-fold cross-validation method involved dividing the data set into five segments. First, doing is the training on segment 1–4, and the 5th part used for validation. In the end, the results assembled, and the accuracy is determined (Table 8).

The results of the proposed scheme were compared with the approach proposed in Abdelhamid et al. (2014), which used MCAC to produce 94.5% accuracy. It was also compared with the method suggested by Barraclough et al. (2013) for phishing detection using neuro-fuzzy which obtained 98.5% accuracy for text-only feature detection. However, in our experiment, we considered this process,

Table 9
ANFIS 5-fold cross-validation with five features input.

Result summary five inputs	Test error		Training error		Training error %	Average Training error%	Training Accuracy %
Feature set 1	0.022388	0.019986	0.015759	0.014006	1.49%		
Feature set 2	0.014926	0.013324	0.015092	0.012795	1.39%		
Feature set 3	0.0346240	0.03662	0.016522	0.013896	1.52%		
Feature set 4	0.014936	0.013324	0.014206	0.016406	1.53%	1.7%	98.3%
Feature set 5	0.020630	0.033310	0.013387	0.012940	1.32%		
Feature set 6	0.0146280	0.02662	0.016828	0.021965	1.49%		
Feature set 7	0.0326210	0.03662	0.021453	0.034447	2.80%		
Average error	0.023897		0.017122				

Table 10
ANFIS validation data source result.

	Image	Text	Frame	Hybrid
2-fold	96.71	93.49	95.44	94.23
5-fold	97.20	98.55	98.06	98.2
10-fold	95.03	91.35	95.1	92.53

but with the fine-tuning of the features arrange together in the same attack pattern for training and testing, and assigning different weight with a reduction in some functions by removing the redundant elements used in their model.

Our result for phishing detection using text only (see Table 4) recorded a small improvement of 0.05% compared to the above conclusion reported by Barraclough et al. (2013). Therefore, we conclude that our method (Table 4) has more accuracy when using text features than the above methods. Our result using image features (see Table 6) only solution recorded 97.2% compared to 96.1% result reported by Afroz and Greenstadt (2009). However, including more elements such as images and frames will further improve the robustness of the system proposed in this study.

7. Conclusion and future work

This paper presented an intelligent phishing detection and protection scheme by employing a new approach using the integrated features of images, frames and text of phishing websites. An efficient ANFIS algorithm was developed, tested and verified for phishing website detection and protection based on the schemes proposed in Aburrous et al. (2010) and Barraclough and Sexton (2015). A set of experiments was performed using 13,000 available datasets. The approach showed an accuracy of 98.3%, which so far, is the best-integrated solutions for web-phishing detection and protection.

The primary contribution of this study is the integration of hybrid features that have been extracted from text, images and frames and that are then used to develop a robust ANFIS solution. Future work will include using another algorithm like deep-learning for phishing web page detection and compare the effectiveness with the current result. More also, a web browser plug-in will be developed based on an efficient algorithm to detect phishing website and thus protect users in real time.

[*Dataset*]

Rami, M., McCluskey, L. and Fadi, T. (2015a) *Phishing Website Dataset*: UCI Machine Learning Repository. Available at: https://archive.ics.uci.edu/ml/machine-learning-databases/00327/Training%20Dataset.arff (Accessed: 12 September 2017).

Rami, M., Thabtah, F. A. and T.L., M. (2015b) *Phishing Websites Dataset*. Available at: http://eprints.hud.ac.uk/24330/9/Mohammad14JulyDS_1.arff (Accessed: 10 September 2017).

References

Abraham, A. (2005). 'Adaptation of fuzzy inference system using neural learning'. In *Fuzzy systems engineering* (pp. 53–83). Berlin, Heidelberg, Springer.

Abunadi, A., Akanbi, O., & Zainal, A. (2013). 'Feature extraction process: A phishing detection approach'. In *13th International Conference on Intelligent Systems Design and Applications in Communications and Network Security (CNS)* (pp. 331–335). 8-10 Dec.IEEE.

Aburrous, M., Hossain, M. A., Dahal, K., & Thabtah, F. (2010). 'Intelligent phishing detection system for e-banking using fuzzy data mining'. *Expert Systems with Applications, 37*(12), 7913–7921.

Aburrous, M., Hossain, M. A., Thabatah, F., & Dahal, K. (2008). Intelligent Phishing Website Detection System using Fuzzy Techniques.

APWG. (2017). Unifying the global response to cybercrime *[Online]*. Washington, D.C: Anti-Phishing Working Group Available at: *https://docs.apwg.org/reports/apwg_trends_report_q2_2016.pdf* .

Arachchilage, N. A. G., Love, S., & Beznosov, K. (2016). 'Phishing threat avoidance behaviour: An empirical investigation'. *Computers in Human Behavior, 60*, 185–197.

Babu, L. D. D., Nirmala, M., & Kumar, K. N. (2010). 'A Survey of Methodologies and Techniques for Detection and Prevention of Phishing Attacks'. *International Journal of Advanced Research in Computer Science, 01*(03), 152–159.

Barraclough, P., & Sexton, G. (2015). 'Phishing website detection fuzzy system modelling'. In *Conference of Science and Information (SAI)* (pp. 1384–1386). 28-30 JulyIEEE.

Barraclough, P. A., Hossain, M. A., Tahir, M. A., Sexton, G., & Aslam, N. (2013). 'Intelligent phishing detection and protection scheme for online transactions. (Report)'. *Expert Systems With Applications, 40*(11), 4697–4706.

Çakıt, E., & Karwowski, W. (2016). 'Predicting the occurrence of adverse events using an adaptive neuro-fuzzy inference system (ANFIS) approach with the help of ANFIS input selection'. *Artificial Intelligence Review, 48*(2), 139–155.

Cohen, A., Nissim, N., Rokach, L., & Elovici, Y. (2016). 'SFEM: Structural Feature Extraction Methodology for the Detection of Malicious Office Documents Using Machine Learning Methods'.

Cruz, A., & Mestrado, N. (2009). 'ANFIS: Adaptive neuro-fuzzy inference systems'. *IM, UFRJ, Mestrado NCE*.

Dariane, A., & Azimi, S. (2016). 'Forecasting streamflow by a combination of a genetic input selection algorithm and wavelet transforms using ANFIS models'. *Journal of Hydrological Sciences, 61*(3), 585–600.

Deshmukh, M., Popat, S. K., & Student, U. (2017). 'Different Techniques for Detection of Phishing Attack'. *International Journal of Engineering Science, 7*(4), 10201–10204.

Elliot, A. J., Maier, M. A., Moller, A. C., Friedman, R., & Meinhardt, J. (2007). 'Color and psychological functioning: The effect of red on performance attainment.'. *Journal of Experimental Psychology, 136*(1), 154–168.

Fatt, J. C. S., Leng, C. K., & Nah, S. S. (2014). 'Phishdentity: Leverage Website Favicon to Offset Polymorphic Phishing Website'. In *Ninth International Conference on Availability, Reliability and Security (ARES)* (pp. 114–119). 8-12 Sept. IEEE.

Financial Fraud Action, U. (2017). *Jan. to Dec. 2016 fraud update payment cards, remote banking and cheque*. United Kingdom, London: Financial Fraud Action, UK Available at: *https://www.financialfraudaction.org.uk/fraudfacts17/* .

Gaunt, R. E. (2016). 'The rate of convergence of some asymptotically chi-square distributed statistics by Stein's method', *arXiv preprint* arXiv:1603.01889.

Getty, D. J., Swets, J. A., Pickett, R. M., & Gonthier, D. (1995). 'System operator response to warnings of danger: A laboratory investigation of the effects of the predictive value of a warning on human response time'. *Journal of Experimental Psychology: Applied, 1*(1), 19–33.

Hripcsak, G., & Rothschild, A. S. (2005). 'Agreement, the f-measure, and reliability in information retrieval'. *Journal of the American Medical Informatics Association, 12*(3), 296–298.

Huang, H., Qian, L., & Wang, Y. (2012). 'An SVM- based technique to detect phishing URLs'. *Journal of Information Technology, 11*(7), 921–925.

Jakobsson, M., & Myers, S. (2006). *Phishing and Countermeasures: understanding the increasing problem of electronic identity theft*. Hoboken, New Jersey: John Wiley & Sons.

Jang, J.-S. R. (1996). 'Input selection for ANFIS learning'. In *Proceedings of the Fifth IEEE international conference on fuzzy systems* (pp. 1493–1499). 11 Sept.Citeseer.

Karaboga, D., & Kaya, E. (2016). 'An adaptive and hybrid artificial bee colony algorithm (ABC) for ANFIS training'. *Applied Soft Computing, 49*, 423–436.

Khadir, R. (2015). 'Efforts and Methodologies used in Phishing Email Detection and

Filtering: A Survey'. *International Journal of Advanced Research in Computer Science, 6*(2), 23–27.
Kim, J., & Kasabov, N. (1999). 'HyFIS: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems'. *Neural Networks, 12*(9), 1301–1319.
Kumar, V., & Kumar, R. (2015). 'Detection of a phishing attack using visual cryptography in ad-hoc network'. In *International Conference on Communications and Signal Processing (ICCSP)* (pp. 1021–1025). 2-4 AprilIEEE.
Lee, J.-L., & Park, D.-H. (2016). 'Phishing Detection Using Web Site Heuristics'. *International Information Institute (Tokyo). Information, 19*(2), 523–530.
Liao, Q. (2008). 'Ransomware: A Growing Threat to SMEs'. *Southwest Decision Science Institutes Annual Conference, Houston, (online)*, 360–366.
Marcolin, C., & Becker, J. (2016). 'Exploring Latent Semantic Analysis in a Big Data (base)'.
Martin, A., Anutthamaa, N., Sathyavathy, M., Francois, M. M. S., & Venkatesan, D. V. P. (2011). 'A framework for predicting phishing websites using neural networks', *arXiv preprint*.
Moghimi, M., & Varjani, A. Y. (2016). 'New rule-based phishing detection method'. *Expert Systems with Applications, 53*, 231–242.
Neil Chou, R. L., Teraguchi, Yuka, Boneh, Dan, & Mitchell, JohnC. (2004). *SpoofGaurd*. San Diego, California Available at: (Accessed: 25 November 2016) https://crypto.stanford.edu/SpoofGuard/.
Office for National Statistics. (2017). *Crime in england and Wales: year ending Dec. 2016*. London, United Kingdom: Office for National Statistics Available at: *https://www.ons.gov.uk/peoplepopulationandcommunity/crimeandjustice/bulletins/crimeinenglandandwales/yearendingjune2016#whats-happening-to-trends-in-fraud* (Accessed: 15 November).
Powers, D. M. (2011). 'Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation'. *8*(3), 153–160.
Rami, M., McCluskey, L., & Fadi, T. (2015a). *Phishing Website Dataset: UCI Machine Learning Repository* Available at: (Accessed: 12 September 2016) https://archive.ics.uci.edu/ml/machine-learning-databases/00327/Training%20Dataset.arff .
Rami, M., Thabtah, F. A., & T. L. , M. (2015b). *Phishing Websites Dataset* Available at: (Accessed: 10 September 2016) http://eprints.hud.ac.uk/24330/9/Mohammad14JulyDS_1.arff.
Shekokar, N. M., Shah, C., Mahajan, M., & Rachh, S. (2015). 'An ideal approach for detection and prevention of phishing attacks'. *Procedia Computer Science, 49*, 82–91.
Stathacopoulou, R., Magoulas, G. D., Grigoriadou, M., & Samarakou, M. (2005). 'Neuro-fuzzy knowledge processing in intelligent learning environments for improved student diagnosis'. *Information Sciences, 170*(2–4), 273–307.
Suied, C., Susini, P., & McAdams, S. (2008). 'Evaluating warning sound urgency with reaction times'. *Journal of Experimental Psychology: Applied, 14*(3), 201–212.
Vidal, R., Ma, Y., & Sastry, S. (2016). 'Generalized Principal Component Analysis'. *Interdisciplinary applied mathematics, 40*, 1–21.
Zareapoor, M., & Seeja, K. (2015). 'Feature Extraction or Feature Selection for Text Classification: A Case Study on Phishing Email Detection'. *International Journal of Information Engineering and Electronic Business, 7*(2), 60–65.
Zeng, Z., Jiang, X., & Neapolitan, R. (2016). 'Discovering causal interactions using Bayesian network scoring and information gain'. *BMC Bioinformatics, 17*(1), 1–4.
Zhang, N., & Yuan, Y. (2013). 'Phishing detection using neural network'. *Department of Computer Science, Department of Statistics, Stanford University* (pp. 1–5).