Evolving Fuzzy Neural Network for Phishing Emails Detection

Article *in* Journal of Computer Science · July 2012

DOI: 10.3844/jcssp.2012.1099.1107

Journal of Computer Science 8 (7): 1099-1107, 2012
ISSN 1549-3636

Evolving Fuzzy Neural Network for Phishing Emails Detection

[1]Ammar ALmomani, [1,2]Tat-Chee Wan,
[1]Altyeb Altaher, [3]Ahmad Manasrah, [2]Eman ALmomani,
[1]Mohammed Anbar, [1]Esraa ALomari and [1]Sureswaran Ramadass
[1]National Advanced IPv6 Centre (NAV6),
[2]School of Computer Sciences,
Universiti Sains Malaysia, 11800 USM, Penang, Malaysia
[3]Faculty of Information Technology and Computer Sciences,
Yarmouk University, 21163, Irbid, Jordan

Abstract: One of the broadly used internet attacks to deceive customers financially in banks and agencies is unknown "zero-day" phishing Emails "zero-day" phishing Emails is a new phishing email that it has not been trained on old dataset, not included in black list. Accordingly, the current paper seeks to Detection and Prediction of unknown "zero-day" phishing Emails by provide a new framework called Phishing Evolving Neural Fuzzy Framework (PENFF) that is based on adoptive Evolving Fuzzy Neural Network (EFuNN). PENFF does the process of detection of phishing email depending on the level of features similarity between body email and URL email features. The totality of the common features vector is controlled by EFuNN to create rules that help predict the phishing email value in online mode. The proposed framework has proved its ability to detect phishing emails by decreasing the error rate in classification process. The current approach is considered a highly compacted framework. As a performance indicator; the Root Mean Square Error (RMSE) and Non-Dimensional Error Index (NDEI) has 0.12 and 0.21 respectively, which has low error rate compared with other approaches Furthermore, this approach has learning capability with footprint consuming memory.

Key words: Phishing emails, Evolving Fuzzy Neural Network (EFuNN), filters email sequentially

INTRODUCTION

Such a type of threats, phishing e-mails, is used to steal sensitive and personal data or user's' account information from their computers. It has become a serious e-crime that financially destructs those organization and agencies that their dealings, contacts, or money transactions with their customers are internet-based all over the world. Terminology speaking, the word phisher denotes the person who sends a phishing message to his/ her target for malicious intentions. Such a message involves an embedded link that takes the users away into a fake website. The latter hunts users gradually through asking those questions about their account information, such as their password or credit card number under the pretext of verifying users' accounts or confirming their billing information (APWG, 2010).

Phishing attacks problems are increasing rapidly. Depending on (McCall, 2007); more than 3.6 million users computer in USA only losses money because of phishing attack. Total losses approximately US $3.2 billion dollars, the number of victims increased from 2.3 million in 2006-3.6 million in 2007. By Symantec Company report SYMANTIC, 2010, more than 95.1 billion phishing messages roughly sent in 2010, about 260 million phishing emails sent daily to customers or financial organizations. eCrime trends report is one of the newest reports depends on the anti phishing work group , which explain that phishing attacks increased 12% yearly. As statistics phishing in Q1 2011 grew 12% over Q2 2010 (IID, 2011).

In this study we try to detect and predict zero-day" phishing Emails by a new framework called phishing Evolving Neural Fuzzy Framework (PENFF) based on adoptive Evolving Fuzzy Neural Networks (EFuNN) (Kasabov and Woodford, 1999), PENFF can be distinguish between phishing email and ham(legitimate) email in online mode footprint consuming memory ,increase the level of accuracy in prediction process and has ability to improve the rule continually, This study

Corresponding Author: Ammar ALmomani, National Advanced IPv6 Centre (NAV6), Universiti Sains Malaysia 11800 USM, Penang, Malaysia

devoted concentration on phishing email only. The remainder of this study is ordered as follows. Sec. 2 related works sec. 3 proposed framework sec.4 materials and methodsresults, sec. 5 conclusions and future study.

MATERIALS AND METHODS

Related works: Phishing emails filtering methods are classified features-based many techniques. The classification can be done via many methods, such as by features extraction, machine learning technique or by clustering methods. Other innovative approaches have been devised for the purpose of detecting phishing e-mails. These approaches are based on the principle of distinguishing between phishing and ham emails. However, email filtering method has been considered one of the practical approaches in detecting phishing e-mail. Its mechanism is based on defining a sender reliance cost by the Domain Name Server (DNS) inquiry and on analyzing message contents. However, this approach is not void of shortages. It, for instance, depends only on the cost of the DNS, which analyzes the address of the sender by the DNS (Inomata *et al.*, 2005). This feature is considered unpractical due to the fact that phishing emails might appear in various shapes and have different features. That is, a phisher might use many techniques other than DNS; a matter that increases the probability of error in detecting such threats.

Resent researcher depends on machine learning technique for detecting phishing emails. There are three types of machine learning usually used in field of phishing email, some of them used supervised learning and some of them used unsupervised learning while some of them used hybrid (supervised/unsupervised) learning technique depend on classifiers .The main rule of the classifiers depend on learning several inputs or features to expect a desirable output.

For detecting phishing emails many approaches have been proposed, features extraction technique supposed by (Fette *et al.*, 2007), his approach called (PILFER) method correlated with machine learning technique depend on features extraction to make a distinguish the phishing email from ham email, Fette used 10 features represent the phishing email features, then by using a random forest algorithm (Cutler *et al.*, 2007) as a classifier to create a number of decision trees, the system able to detect the type of new email have the same style of features. The accuracy in this framework has more than 96%, with false positive rate 0.1 and 4% false negative rate. However, this technique depend on supervised learning so it is weak in detect "zero-day" attack because it need training on the same attack to expect the new one.

(Abu-Nimeh *et al.*, 2007) compared six classifiers related with machine learning technique for phishing email prediction. The author suggests that the use of cost-sensitive measures penalizes the false positives nine times more than the false negatives; the results indicated that there is no standard classifier for phishing email prediction (Abu-Nimeh *et al.*, 2007).

Another widely deployed technique used multi classifier related with machine learning for phishing email detection is (Saberi *et al.*, 2007), the proposed method accuracy detected 94.4% of phishing emails. An another approach depend on three tier classification to detect phishing emails is (Islam *et al.*, 2009), if the first two classifier can't classify well the final tier will have the final decision , the average accuracy of this approach reach up to 97%. However, this technique is characterized by lengthy time consumption and complexity of analysis since this technique requires many stages before arriving at the final decision. A new method based on machine learning depend on profiling of phishing email, (Yearwood *et al.*, 2010), this method classify seven input features as binary value, 0 or 1, while 1 denoted to include this feature, 0 otherwise.

Other approaches used clustering method in phishing email detection method, Mori (Dazeley *et al.*, 2010), his proposed depends on shared method between unsupervised clustering algorithms with supervised classification algorithms then train the data by Consensus Clustering. (Basnet *et al*., 2008) is one of a new researcher employed k-means algorithm for phishing email detection, this approach has accuracy 90.7,working with unsupervised learning technique and he selected sixteen features of phishing email, the limitation of this algorithm it work on offline mode and the accuracy level still low. For all of these reason we still need a new technique able to detect and predict "zero-day" phishing Emails.

Proposed framework section: Our proposed framework explained clearly in Fig. 1 which provides order steps of how to distinguish between phishing emails and ham emails. This framework divided to four stages, first stage pre-processing of the data set, second stage is email object similarity and third stage is integrated with Evolving Fuzzy Neural Networks (Kasabov and Woodford, 1999) to build PENF, for detection and prediction phishing emails in online mode. All of this stages will work after determine the features of phishing email which used in our framework.

Phishing Email Features used in PENFF: PENFF separately collects and filters email sequentially, which depends on sixteen features represent the most effective features of phishing email.

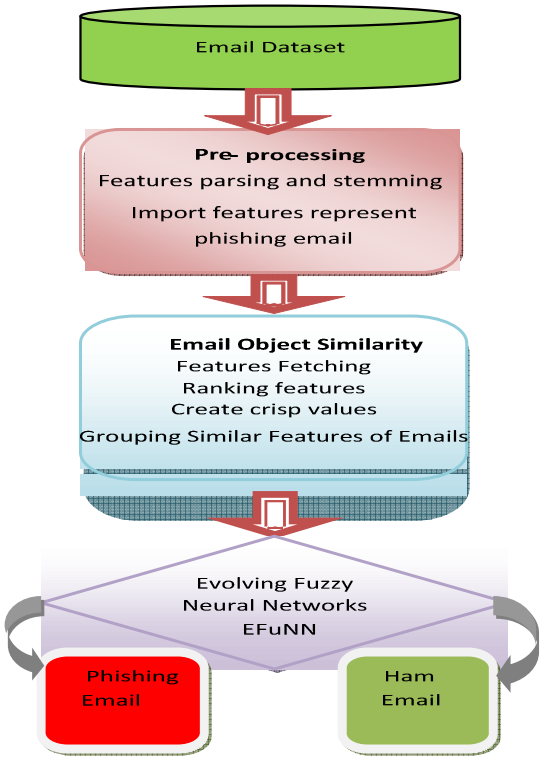

Fig. 1: The overall Phishing Evolving Neural Fuzzy Framework- PENFF

All of these features proposed in earlier (Christine *et al*., 2004; Fette *et al.*, 2007), with some enhancements on features extractions technique. The sixteen features represented as binary value (0, 1),"1" to include the features and "0" otherwise. The sixteen features explained as follow:

Using IP address (ipaddress): A number of phishers depend on their PCs as hosts for a phishing Web site. However, these PCs sometimes do not have DNS entries. Therefore, the easiest way to hide the normal form of a URL is to use IP addresses. Legitimate companies rarely use an IP address as a link page. We take "http://218.56.77.130/paypal.com" as example. For this feature, if an e-mail message has a link similar to an IP address, the probability of the e-mail being a phishing e-mail is increased. This is a binary feature that takes a value of 1 if the e-mail contains a URL similar to an IP address and 0 otherwise.

Difference between sender domains with the domain of embedded links (diffsenlindom): When the link embedded in the HTML does not equal the sender's domain, it is most likely a phishing e-mail. For example, an e-mail may contain the following information: From: "identdep_op720@southtrust.com", URL link: "http://accounts.keybank.com". This is a binary feature. Therefore, if the domain name in the "from "field does not equal the domain name in the URL (embedded HTML), the value of this feature is 1 and 0 otherwise.

Number of links (nulinks): One of the features of a phishing e-mail is a number of links embedded in HTML parts. In the proposed framework, links are distinguished based on tags <*a*> with *HREF*. This feature includes "mail to:" links. After analyzing the data set, we suggest this binary feature takes a value of 1 if there are more than three embedded links and 0 otherwise.

Nonmatching between target and text of URLs (Tardiflink): If they have different host a value "1" and"0" otherwise.

Number of different domains (nudiffdomain): The main part of a domain name which starts with http:// or https:// is extracted for all URLs starting with http:// or https://. In the present study, the main part of a link is assumed to include the section after the first dot up to the first slash ("/") if the link has a long domain name. For example, the "main" part of "www.sg.echool.edu" is sg.echool.edu and the "main" part of "www.jordan.com" is jordan.com. After this process, the number of different domains is calculated. After analyzing 4,000 phishing and ham e-mails, many phishing e-mails were found to have more than three domains. Therefore, we suggest this feature takes a value of 1 if the number of different domains is more than three and 0 otherwise.

Number of dots in a domain (nodot): Attackers utilize many methods to stage a phishing attack. One method depends on the inclusion of a sub-domain. We take "http://www.may-bank.update.data.com" as example. This link appears to be hosted by Maybank, but it is actually not. There are four dots in the domain. Generally, a legitimate company will have no more than three dots on its domain name. Therefore, this feature depends on determining the maximum number of dots. We suggest it takes a value of 1 if there are more than three dots in the domain and 0 otherwise.

Click here (clickhere): Many phishers use words like "click here," "click," or "here" in the text portion of their links in order to hide a suspicious domain name. When users click on such words, they are redirected to a phishing Web site. We take <a HREF="http://61.119.228.47/.eBay/" > click here </A>

as example. If an e-mail message has one of the three words mentioned above, it is flagged as a phishing e-mail takes a value of 1 and 0 otherwise.

Pictures used as links (NoPicLinks): Some attackers use an image as a link to hide fraudulent URLs. We take <img src="http://www.paypalobjects.com/en_US" as example. The maximum number of images used as a link is calculated based on the tag "<img src=URL" embedded in the HTML. After analysis, we suggest this is binary feature takes a value of 1 if there are more than two pictures used as links and 0 otherwise.

HTML e-mail (html e-mail): At present, creating a phishing e-mail is difficult without using an HTML code because an HTML code enables an embedded link to connect directly to other Web sites. The presence of an HTML code in an e-mail can be determined using MIME types. If the MIME type is either text/html or a multipart/alternative, an HTML code is embedded in the message. This is a binary feature takes a value of 1 if no HTML code is embedded and 0 otherwise.

Use of JavaScript (jascript): One of the primary methods used by an attacker to build a phishing e-mail is the use of java script because with this simple language, the phisher can program pop-up windows. The phisher can then change the status bar of a Web browser, enabling him/her to build a complex attack using an embedded script code inside a link. An e-mail message can be determined to have a java script by the tag "JavaScript" or <*script*>. This binary feature takes a value of 1 if the message has a java script code and 0 otherwise.

Non-standard port in the URL (nonstport): A server accesses Web pages using ports and a few phishers use non-standard ports to hide their identity and location. Web pages use port 80 as default and some normal ports such as 443 are used by legitimate companies. The port number in a URL link comes after a colon. For example, in http://www.paybankonline.com:ac@50.28.170.70:8030 /,:8030 represents the port number. This is a binary feature that takes a value of 1 if the e-mail message uses a port other than 80 or 443 and 0 otherwise.

URL containing hexadecimal characters or @ symbol (hexorat): Some attackers use hexadecimal character codes to hide embedded URLs. Attackers can write an IP address using the "%" symbol to build a hexadecimal number. Sometimes, they use the "@" symbol to confuse users. This binary feature takes a value of 1 if the message URL contains either the "%"or @ symbol and 0 otherwise.

Message size (messize): Message size refers to the size of an e-mail in bytes. Most phishing e-mails have a size of less than 25 kb. However, based on a semantic report SYMANTIC, 2010, more than 90% of phishing e-mails have a size of less than 20 kb. Therefore, we suggest this binary feature takes a value of 1 if the message size is less than 25 kb and 0 otherwise.

Faking a secure connection (facksecon): One of the most fraudulent applications used by phishers utilizes URLs that begin with "https://" (instead of using "http://") to trick users into believing that the link is a legitimate URL supported by a Secure Sockets Layer certificate. We take https://www.maybank.com%01 [string of ~ 60 ―%01 elided]@203.172.185.20/f/ as example. Clicking on this link will redirect the user to "http:// 203.172.185.20/f/" which tries to mimic a secure connection. This binary feature takes a value of 1 if the embedded URL starts with https:// and 0 otherwise

HTML form (htmlform): One of the earliest features used to collect user information directly by e-mail utilizes the FORM feature. This feature is a simple code written using HTML, which allows a form requiring the entry of user information such as usernames and passwords to be built. The FORM feature uses a button to submit this information to a phisher account. This feature can be detected if the HTML code has a <FORM>tag. This binary feature takes a value of 1 if the message has a <form> tag and 0 otherwise.

Spam features (spamfeatures): More than 90% of daily e-mails are classified as spam. Phishing e-mails comprise a subset of spam. One of the most powerful software tools available for free and is from an open source, which can classify e-mails as either spam or not, is SpamAssassin (pop3proxy, 2010). In the current study, SpamAssassin version 3.2.3.5 was used with the default rule and using the threshold, more than 40 Boolean features were examined by the software. This binary feature takes a value of 1 if the message is classified as spam and 0 otherwise 5.

Pre-processing: Pre-processing has two parts, firstly parsing and stemming email. Parsing process used to extract features of phishing emails. Stemming process used to clear the text data included in phishing email features. Secondly is importing features which represent phishing email, then convert the data to binary values (1, 0), while "1" act included phishing email feature, "0" otherwise. Then take the processed email dataset to Email Object Similarity, the extraction depends on writing series of java script code to extract the sixteen features discussed before.

Email object similarity: Email Object Similarity consists of three processes after fetching the binary data which symbolize each email in the data set, as follow.

Feature ranking and classification: The most effective algorithm which used to know Features ranking and classification is Information Gain Ratio method (IGR). This method working based on extraction the similarity between set of emails, then gives the high weight to most efficient features. This method evaluates the most effective feature with the class of phishing emails and ham (legitimate) emails based on IGR (Mori, 2002), This algorithm Evaluates the best features with respect to the class which calculated as follows. The Information Gain Ratio (IGR), gain_r(X, C) of feature X in class C is calculated as follows Eq. 1:

$$\begin{aligned}
\text{Gain_r(X, C)} &= \text{gain (X, C)/split_info(C)}\\
\text{Gain(X, C)} &= \text{entropy(X, C)-entropy}_p\text{(X, C)}\\
\text{Entropy(X, C)} &= -p(X/C)\log_2 p(X/C))\\
&\quad - (1-p(X/C))\log 2p(X/C))\log 2\ (1-p(X/C)),\\
P(X/C) &= \text{freq(C, X)/|C|},\\
\text{Entropy}_P\text{(X, C)} &= Ci\ /|C|)\ \text{entropy(X,Ci)},\\
\text{Split_info(C)} &= -\sum_i \left(\frac{C_i}{C}\right) log|\ (Ci|/i|C|), \qquad (1)
\end{aligned}$$

With notice, freq (X, C), C_i and $|C_i|$ denoted to the frequency of features X in class C, the *i*-th sub-class of C and the number of features in Ci, respectively. For this algorithm we used open source software Weka for Feature ranking and classification Waikato, 2010.

Table 1 shows the phishing e-mail features with The Information Gain Ratio (IGR) value related with each features, with note that, "html e-mail" is the feature with the top quality, where as "diffsenlindom" is the slightest helpful and possibly causes noise in the classifier. This value take after analyzing the full dataset wich explain in part 4.

Create crisp values: Fuzzy Logic (FL) is part of our framework, working with crisp value, therefore for creation of crisp value we Converts the binary value (0, 1) of all emails dataset to crisp value by division all features on 100 score based on algorithm 2. then nearest value to the largest integer number Eq. 2:

$$y_i = (100*IGR_i/\text{sum (IGRi)}) \qquad (2)$$

(ALmomani, 2011) Where y is the crisp value of "i" feature and IGR is (Information Gain ratio). Then for each feature in the dataset we multiply each binary value with it is result of crisp value. As you show in Table 2.

Table 1: Phishing e-mail features with IGR

Features number (i)	Features (privations)	Information Gain Ratio (IGR)
1	Htmlemail	0.59503
2	Facksecon	0.31765
3	Tardiflink	0.31685
4	NoPicLink	0.25802
5	Ipaddress	0.23858
6	Clickhere	0.22802
7	Nudiffdomain	0.22105
8	Nodot	0.13793
9	Hexorat	0.10658
10	Nulinks	0.17158
11	Htmlform	0.01486
12	Spam features	0.3215
13	Messize	0.02702
14	Nonstport	0.02132
15	Jascript	0.0176
16	Diffsenlindom	0.00493
	Sum	2.99852

Table 2: Phishing e-mail groups features with crisp values

Crisp Value (y)	Group names of features	Features (privations)
18	BODYEMAIL	Htmlemail
9	features	NoPicLink
7		Nudiffdomain
6		Nulinks
1		Htmlform
11		Spamfeatures
1		Messize
1		Jascript
11	URL features	Facksecon
11		Tardiflink
8		Ipaddress
8		Clickhere
5		Nodot
4		Hexorat
1		Nonstport
1		Diffsenlindom
Sum=100		

Grouping similar features of emails: For the purpose of gaining an easy and fast method for features classification, Grouping Similar Features of emails method has been proposed in the present study. This method consists of two groups: the body features group and URL features group. Each group sums eight features from the other group so the total will be sixteen features distributed within two columns and stored in database to be used later in the proposed framework, PENFF. Consider Table 2.

Adoptive Evolving Fuzzy Neural Networks (EFuNN): The expert systems EFuNN, is an intelligent system that evolves in accordance with the Evolving Connectionist System (ECOS) (Kasabov and Woodford, 1999). This system, like traditional expert systems with more power full included working with unfixed number of rules used to develop the Artificial Intelligent (AI). The system, EFuNN, consists of five layers.

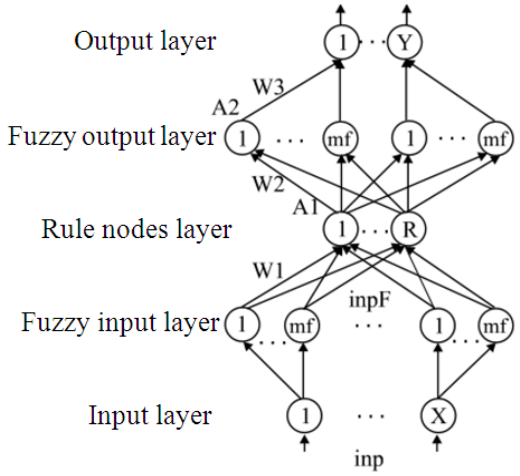

Fig. 2: EFuNN's architecture

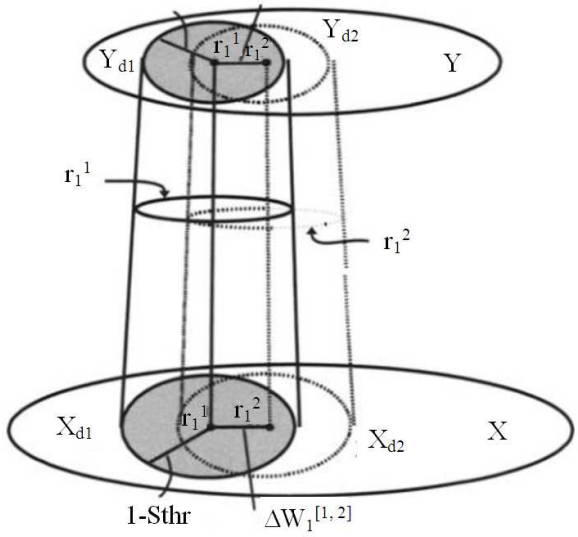

Fig. 3: EFuNN with evolving hyper-sphere technique to build new rule

It is flexible with respect to the dynamic rule, works in online mode and interacts with dynamically changing environment. Such a system can solve the complexity and changeability of many real world problems. It grows throughout the process of working and adopts many techniques, such as that of the Neural Network technique (NN). However, the only difference between EFuNN and NN is that all nodes in the former are created during the process of learning.

During the process of learning, Membership Function node (MF) that is characterized by being fuzzy label neurons is modified. Two neurons values can symbolize fuzzy values of different sizes, such as: _small‘,‘ medium‘ and _large‘, Moreover, (MF) can interact with these neurons (triangular, Gaussian, etc.,). Moreover, it is to be noted that new neurons can be evolved in this layer together with input vectors.

If the variable value does not belong to any of the existing (MF), this means that the value is greater than that of the membership threshold. Accordingly, the system will create a new fuzzy input neuron by adopting stage of *EFuNN*.

EFuNNs algorithm: To evolve *EFuNN*, a new rule node (rn) is created. The following function shows the input and output connection weights: W1 (rn) = EX; W2 (rn) = TE, where TE is the fuzzy output vector for the current fuzzy input vector EX.

Usually, “one-of-n” EFuNN, the highest activation of the rule node is sent to the next level. The linear functions, on the other hand, are used as an activation function for the fuzzy output neurons.

As for the “many-of-n” state, all the activation values of rule nodes that are above the activation threshold (Athr) will further be sent to the connectionist structure.

For more illustration, Fig. 2, which explains the five-layer structures of EFuNNs that are fuzzy neural networks used for the purpose of detection and prediction (Koprinska and Kasabov, 1999).

EFuNN in the detection and prediction of phishing emails: EFuNN used to generate evolving fuzzy rules. The functions depend on learning data continuously. The third layer of EFuNN consists of the rule nodes, which evolve through either supervised or unsupervised learning. This layer represents prototypes of input and output data associations. For more clarification, consider Fig. 2, which explicates graphically the rule created during the evolving process associates a hyper-sphere from the fuzzy input space to a hyper-sphere from the fuzzy output space.

It is to be noted that w1(r) and w2(r) represent two vectors of features of phishing emails that are used for weights connection. The system adjustment of the latter is done during the supervised learning phase where as that of the former is done during the unsupervised learning phase. The process of adjustment of the w2(r) depends on output error where as that of the w1(r) is based on calculating the similarity within a local area of the problem space.

Normalizing the fuzzy difference between X_{d1} and X^{d2} is somehow less than the radius r Normalization of fuzzy difference between Y_{dl} and Y_{d2}. This is because the process of normalizing the fuzzy difference is less than the error threshold (Err_{thr}). A rule node r1 with center r_1^1 and the new data point $d_2 = (x_{d2}, y_{d2})$ that is within the shaded area are clearly indicated in Fig. 3.

Through the associative (learning), new data point will be implemented to a rule node by the features vector of phishing emails. The center of this node, which is called the 'cluster center' is adjusted in the fuzzy input space that is based on a learning rate l_{r1} while the fuzzy of output space is based on a learning rate, Ir_2 of two data points. The center of cluster r_1^1 is updated to the new position" to become r_1^2. This step can be implemented mathematically by changing in the connection weight of the rule node r_1 from $W_I(r_1^1)$ and $W^2(r_1^1)$ to $W_I(r_1^2)$ And $W_2(r_1^2)$ as follow Eq. 3 and 4:

$$W_1(r_1^2) = W_1(r_1^1) + lr_1 * Ds(X_{d1}, X_{d2}) \quad (3)$$

$$W_1(r_1^2) = W_2(r_1^1) + lr_1 * Err(Y_{d1}, Y_{d2}) * A_1(R_1^1) \quad (4)$$

Where $Err(Y_{d1}, Y_{d2}) = Ds(Y_{dI}, Y_{d2}) = Y_{d1}-Y_{d2}$ is the signed value instead of the absolute value of difference vector. $A_I(r_l^I)$ is the activation of the rule node r_l^I for the input vector X_{d2}. As you shown in Fig. 3 (Kasabov, 2007).

RESULTS AND DISCUSSION

Implementation and test result: The implementation will be achieved by using Matlab version 7.10 (Diller, 2010). On PC has 3GB of Memory (RAM) and Duo CPU 2.66 GHz. From the data set, the two classes training and testing based on 10-Fold Cross Validation by randomly Method.

The first class consists of 2000 phishing emails that are offered by the monkey website. The second one, on the other hand, consists of 2000 ham emails taken from the spamassassin (LLCB, 2010).

As a performance indicator of the framework to know the accurate prediction results related with "zero-day" phishing email. The current study uses Root Mean Square Error (RMSE) as shown in Eq. 5. RMSE = 0 means that the model output exactly matches the observed output.

$$RMSE = \sqrt{\sum_{i=1}^{N} \frac{(yi - \hat{yi})^2}{N}} \quad (5)$$

Where n is the number of emails input samples, y_i is the i_{th} actual output, $\hat{y}_i$ the ith framework output. The Non-Dimensional Error Index (NDEI) is used; it is known as the ratio of the Root Mean Square Error (RMSE). It divides the standard deviation of the target data std (y (t)), as it is shown in Eq. 6 (Kim and Kasabov, 1999).

$$NDEI = \frac{RMSE}{std(y(t))} \quad (6)$$

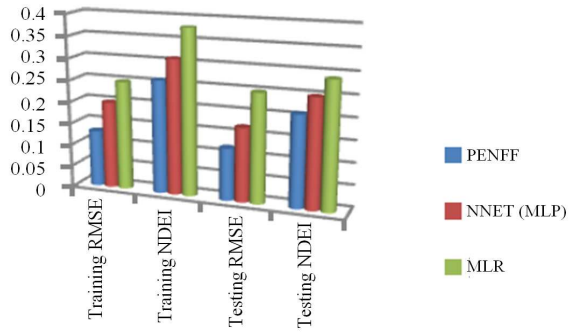

Fig. 4: Comparison between PENFF with other methods

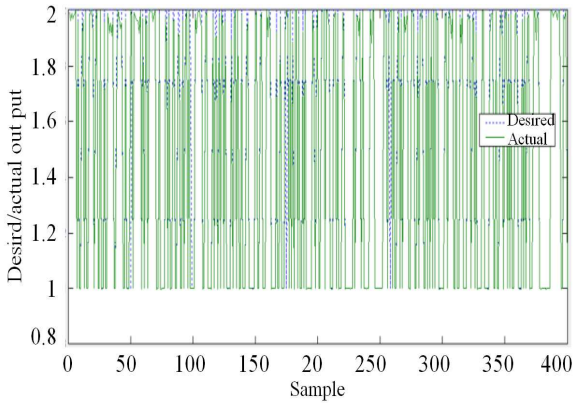

Fig. 5: PENFF–Accuracy level in testing sample

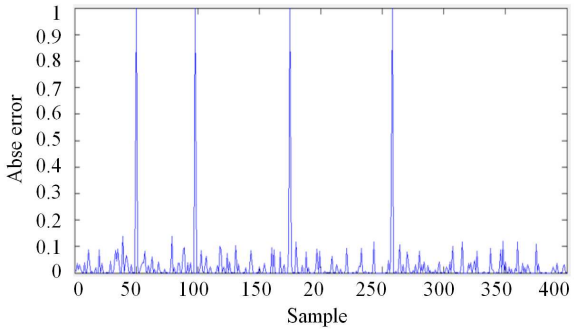

Fig. 6: PENFF–Error rate-Testing s

Table 3: PENFF for phishing email prediction in learning and testing phase compared with other methods

Method	PENFF	NNET (MLP)	MLR
Learning methods	Supervised /unsupervised	Supervised	Supervised
Training RMSE	0.13	0.20	0.25
Training NDEI	0.26	0.31	0.38
Testing RMSE	0.12	0.17	0.25
Testing NDEI	0.21	0.25	0.29

For each experiment we learn 9 folds then test the 10th fold. We used different thresholds and different weights then we take the average of results from all experiments

as shown in Table 3 .The rule will be extracted from the learning dataset to be used later in the testing stage. We used our framework in 3 algorithms to compare the effectiveness of EufNN compared with Neural Network (NNet) and Multiple Linear Regression (MLR) algorithms which is usually used to detect phishing email attack.

For more illustration, see Table 3and Fig. 4. From the results appeared in Table 3 and Fig. 4, the error rate of the NDEI and RMSE in our framework (PENFF) has the lowest values compared with other methods. Accordingly, this technique proves the ability to have more accurate prediction result with. Fig. 5 shows the accuracy level in testing of 400 samples and Fig. 6 show PENFF–Error rate-Testing sample in 2d space.

Form Fig. 5and 6 respectively we can see the high level of accuracy in prediction process and detect the "zer-day" phishing email attack and the level of error rate was so low, mostly less than 0.1.

CONCLUSION

Unknown "Zero-day" phishing email still one of the biggest problems in machine learning to detect phishing email attack.

PENFF proved the ability to distinguish between phishing emails and ham emails in online mode, Depend on new technique based on binary value 0 or 1 for all used features, 1 denoted as phishing flag features, "0" otherwise. PENFF built by taking the advantages EFuNN. PENFF has many power full features which usually used for online system, incrementally; our framework result proved the ability to have more accuracy than other approaches with ability to implement in life-long learning systems. For the future work we suggest to use more dynamic system to build system able to work in real implementations, to have more accuracy with high performance.

ACKNOWLEDGMENT

This research is supported by National Advanced IPv6 Centre of Excellence (NAV6) University Sains Malaysia (USM), under grant: No:1001/PNAV/857001.

REFERENCES

Abu-Nimeh, S., D. Nappa, X. Wang and S. Nair, 2007. A comparison of machine learning techniques for phishing detection. Proceedings of the Anti-Phishing Working Groups 2nd Annual Ecrime Researchers Summit, Oct. 22-24, ACM, Pittsburgh, USA, pp: 60-69. DOI: 10.1145/1299015.1299021

ALmomani, A., 2011. An online model on evolving phishing e-mail detection and classification method. J. Applied Sci., 11: 3301-3307.

APWG, 2010. Phishing activity trends report. APWG.

Basnet, R., S. Mukkamala and A.H. Sung, 2008. Detection of phishing attacks: A machine learning approach. Studies Fuzziness Soft Comput., 226: 373-383.

Christine, E., J.J.O. Drake and J. Eugene and Koontz, 2004. Anatomy of a phishing email. Proceedings of the 1st Conference on Email and Anti-Spam, (CEAS' 04), Mountain View, CA, USA.

Cutler, D.R., T.C. Edwards Jr., K.H. Beard, A. Cutler and K.T. Hess *et al*., 2007. Random forests for classification in ecology. Ecology, 88: 2783-2792. PMID: 18051647

Dazeley, R., J.L. Yearwood, B.H. Kang and A.V. Kelarev, 2010. Consensus clustering and supervised classification for profiling phishing emails in internet commerce security. Proceedings of the 11th International Conference on Knowledge Management and Acquisition for Smart Systems and Services, (PKAW'10), Heidelberg, pp: 235-246.

Diller, D., 2010. Math Work Stations: Independent Learning You Can Count On, K-2. 1st Edn., Stenhouse Publishers, Portland, ISBN-10: 1571107932, pp: 299.

Fette, I., N. Sadeh and A. Tomasic, 2007. Learning to detect phishing emails. Proceedings of the 16th International World Wide Web Conference, May 08-12, ACM Press, Banff, Alberta, Canada, pp: 649-656. DOI: 10.1145/1242572.1242660

IID, 2011. Q3 2011 eCrime Trends Report released. IID.

Inomata, A., M. Rahman, T. Okamoto and E. Okamoto, 2005. A novel mail filtering method against phishing. Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Aug. 24-26, IEEE Xplore Press, Japan, pp: 221-224. DOI: 10.1109/PACRIM.2005.1517265

Islam, M.R., J. Abawajy and M. Warren, 2009. Multi-tier phishing email classification with an impact of classifier rescheduling. Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms and Networks, Dec. 14-16, IEEE Xplore Press, USA., pp: 789-793. DOI: 10.1109/I-SPAN.2009.142

Kasabov, N. and B. Woodford, 1999. Rule insertion and rule extraction from evolving fuzzy neural networks: Algorithms and applications for building adaptive, intelligent expert systems. Proceedings of the IEEE International Fuzzy Systems Conference, Aug. 22-25, IEEE Xplore Press, New Zealand, pp: 1406-1411. DOI: 10.1109/FUZZY.1999.790109

Kasabov, N.K., 2007. Evolving Connectionist Systems: The Knowledge Engineering Approach. 1st Edn., Springer, London, ISBN-10: 1846283450, pp: 457.

Kim, J. and N. Kasabov, 1999. Hyfis: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems. Neural Netw., 12: 1301-1319. PMID: 12662634

Koprinska, I. and N. Kasabov, 1999. An application of evolving fuzzy neural network for compressed video parsing. Proceedings of the ICONIP/ANZIIS/ANNES'99 Workshop, Nov. 22-24, Dunedin, New Zealand, pp: 96-102.

LLCB, 2010. Apache Software Foundation Projects: Apache Httpd Modules, List of Apache Software Foundation Projects, Mod Ssl, Apache Gump, Mod Qos. 1st Edn., General Books LLC., ISBN-10: 1157769128, pp: 76.

McCall, T., 2007. Gartner survey shows phishing attacks escalated in 2007; more than \$3 billion lost to these attacks. Stephane GALLAND.

Mori, T., 2002. Information gain ratio as term weight: The case of summarization of IR results. Proceedings of the 19th International Conference on Computational Linguistics, (CL' 02), ACLS, USA., pp: 1-7. DOI: 10.3115/1072228.1072246

Saberi, A., M. Vahidi and B.M. Bidgoli, 2007. Learn to detect phishing scams using learning and ensemble? methods. Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops, (WIIATW' 07), ACM, USA., pp: 311-314.

Yearwood, J., M. Mammadov and A. Banerjee, 2010. Profiling phishing emails based on hyperlink information. Proceedings of the International Conference on Advances in Social Networks Analysis and Mining, Aug. 9-11, IEEE Xplore Press, USA., pp: 120-127. DOI: 10.1109/ASONAM.2010.56