An Efficient Approach for Phishing Detection Using Neuro-Fuzzy Model

Luong Anh Tuan Nguyen, Ba Lam To, and Huu Khuong Nguyen
Ho Chi Minh City University of Transport, Vietnam
Email: {nlatuan, nhkhuong}@hcmutrans.edu.vn, tblam83@gmail.com

***Abstract*—Nowadays, online transactions are becoming more and more popular in modern society. As a result, Phishing is an attempt by an individual or a group of people to steal personal information such as password, banking account and credit card information, etc. Most of these phishing web pages look similar to the real web pages in terms of website interface and uniform resource locator (URL) address. Many techniques have been proposed to detect phishing websites, such as Blacklist-based technique, Heuristic-based technique, etc. However, the numbers of victims have been increasing due to inefficient protection technique. Neural networks and fuzzy systems can be combined to join its advantages and to cure its individual illness. This paper proposed a new neuro-fuzzy model without using rule sets for phishing detection. Specifically, the proposed technique calculates the value of heuristics from membership functions. Then, the weights are generated by a neural network. The proposed technique is evaluated with the datasets of 11,660 phishing sites and 10,000 legitimate sites. The results show that the proposed technique can detect over 99% phishing sites.**

***Index Terms*—phishing, neuro-fuzzy, neural network**

I. INTRODUCTION

Phishers use a number of techniques to lure their victims, including email messages, instant messages, forum posts, phone calls, and text messages. With these activities of phishing, it causes severe economy loss all over the world. APWG's second half report for 2010 claimed that phishing attacks grew 142% over the first half of 2010. The report also classifies the targets as comprising 37.9% payment services, 33.1% financial institutions, 6.6% classified, 4.6% gaming, 2.8% social networks, and the remainder in other categories. In 2011, 83% of Americans and 85% of Europeans regularly shopped online (Fortune Magazine, 2011). Meanwhile, phishing sites are also growing rapidly in quality and quantity. Therefore, the risk of stealing user information is extremely high. Because of these reasons, detecting phishing problem is very urgent, complex and extremely important problem in modern society. Recently, there have been many studies which against phishing based on the characteristics of site, such as URL of website, content of website, combining both the website URL and content, source code of website or screenshot of website, etc. However, each of study has its own strengths and weaknesses. There is still not a sufficient method. In this paper, a new approach is proposed to detect the phishing sites that focuses on the features of URL (PrimaryDomain, SubDomain, PathDomain) and the ranking of site (PageRank, AlexaRank, AlexaReputation). Then, a proposed neuro-fuzzy network is a system which reduces the error and increases the performance. The proposed neuro-fuzzy model uses computational models to perform without rule sets. The proposed solution achieved detection accuracy above 99% with low false signals. The rest of this paper is organized as follows: Section II presents the related works. System design is shown in section III. Section IV evaluates the accuracy of the method. Finally, Section V concludes the paper and figures out the future works.

II. RELATED WORKS

The phishing detection techniques are classified into three categories such as blacklist, heuristic and machine learning. In the first approach, the phishing detection technique [1]-[4] maintains a list of phishing websites called blacklist. However, the blacklist technique is inefficient due to the rapid growth in the number of phishing sites. Therefore, the heuristic and machine learning approaches have received more attraction of researchers. Cantina [5] presented the algorithm TF-IDF based on 27 features of webpage. This technique can detect 97% phishing sites with 6% false positives. Although this technique is efficient, the time extracting 27 features of webpage is too long to meet real time demand and some features are not necessary for improving the phishing detection accuracy. Moreover, the evaluation dataset is quite small. Similarly, Cantina+ [6] used machine learning techniques based on 15 features of webpage and only six of 15 features are efficient for phishing detection such as bad form, Bad action fields, Non-matching URLs, Page in top search results, Search copyright brand plus domain and Search copyright brand plus hostname. In [7], the author used the URL to detect phishing sites automatically by extracting and verifying different terms of a URL through search engine. Even though this paper proposed a new interesting technique, the detection rate is quite low (54.3%). The technique [8] developed a content-based approach to detect phishing called CANTINA which considers the Google PageRank value of a page; however, the evaluation dataset is quite

Manuscript received September 21, 2014; revised December 11, 2014.

doi: 10.12720/joace.3.6.519-525

small. The characteristic of the source code is used to detect phishing sites in [9]. The authors in [10] have proposed fuzzy technique based on 27 features of webpage, classified into 3 layers. Each feature has three linguistic values: low, moderate, high. The fuzzy technique has built a rule set, triangular and trapezoidal membership functions. The achieved website phishing rate of the technique is 86.2%. However, there exist many drawbacks in [10]. First, the rule sets are not objective and greatly depend on the builder. Second, the weight of each main criterion is used without any clarification. Finally, the proposed heuristics are not optimal and really effective. The authors [11] have proposed neural network technique. The technique [11] had been built 3 layers including the input layer, the hidden layer and the output layer. The best achieved rate of the technique is 95%. However, there exist some drawbacks in [11]. First, a number of hidden nodes and activation function must be determined through experimentation. Second, the authors do not explain why using one hidden layer. Third, the value of features is calculated without any clarification. Finally, the datasets are not big enough.

In the previous techniques, the URL has a minor role in detecting phishing websites. In this paper, we focus on URL features and apply the neuro-fuzzy technique to detect phishing sites. The contribution of our paper is the following: i) The new heuristics have been proposed to detect phishing website more effectively and rapidly. ii) The threshold values used in the membership functions are derived from the big data set so that the model is still equivalent for the new data set. iii) The weights of heuristic are more optimize because the weights are trained by neural network. iv). The rule sets are not utilized. Hence, the result will be more precise and objective.

III. System Design

A. Neuro-Fuzzy Network Without Rule Set

Neural networks and fuzzy logic, which are termed soft computing techniques, are tools of establishing intelligent systems. A fuzzy inference system (FIS) employing fuzzy if-then rules in acquiring knowledge from human experts can deal with imprecise and vague problems [12]. FISs have been widely used in many applications including optimization, control, and system identification. Fuzzy systems do not usually learn and adjust themselves [13], whereas a neural network (NN) has the capacity to learn from its environment, self-organize, and adapt in an interactive way. Because of these reasons, a neuro-fuzzy system, which is the combination of fuzzy system and neural network, has been introduced to produce a complete fuzzy-rule-based system [14], [15]. However, the rule sets are not objective and greatly depend on the builder, so the rule sets are not utilized in the proposed neuro-fuzzy model. Hence, the result will be more precise and objective.

B. URL

A URL (Uniform Resource Locator) is used to locate the resources [16]. The structure of URL is as follows:

< protocol > : // < subdomain > . <primarydomain > . < TLD > / < pathdomain >

For example, with the URL:

http://www.paypal.abc.net/login/index.html

There are six components as follows: Protocol is http, Subdomain is paypal, Primarydomain is abc, TLD is net, Domain is abc.net, Pathdomain is login/index.html

C. Feature of URL

Phishers usually try to make the Internet address (URL) of phishing sites look similar to legitimate sites to fool online users. They can not use the exact URL of the legitimate site, they make more spelling mistake the features of URL such as PrimaryDomain, SubDomain, PathDomain. For example, the URL www.applle.com looks similar to well known website www.apple.com, if users are not careful, they will think that they are on the “apple” site.

D. Feature of Domain’s Ranking

It is obvious that the phishing sites are neither accessed by the users nor linked by the other websites. Therefore, the ranking of site such as PageRank, AlexaRank, AlexaReputation can also help to detect phishing sites. Phishers usually make fake-site of famous site, but the ranking of fake-site is not high. We can also use the rankings to classify whether a site is phishing site.

E. System Model Design

The model can be depicted in Fig. 1.

- *Phase I* – Selecting four features of URL: Four features are extracted from URL such as Domain, PrimaryDomain, SubDomain and PathDomain.
- *Phase II* - Calculating six values of the heuristics: Six values of the heuristics are calculated, six heuristics are six input node of the neuro-fuzzy network.
- *Phase III* – Neuro-fuzzy Network: The neuro-fuzzy network performs to calculate the value of the output node.
- *Phase IV* - Classifying the websites: We based on the output value of the output node to decide whether a website is a phishing website.

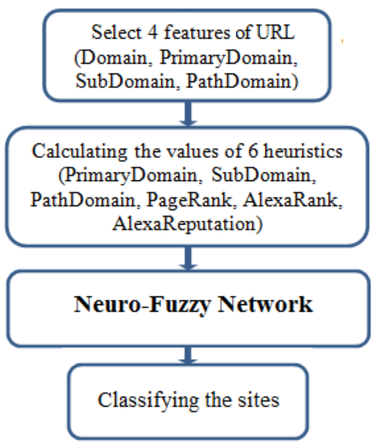

Figure 1. System model

F. Neuro-Fuzzy Network Model

1) The model

The proposed neuro-fuzzy network model was designed as in Fig. 2.

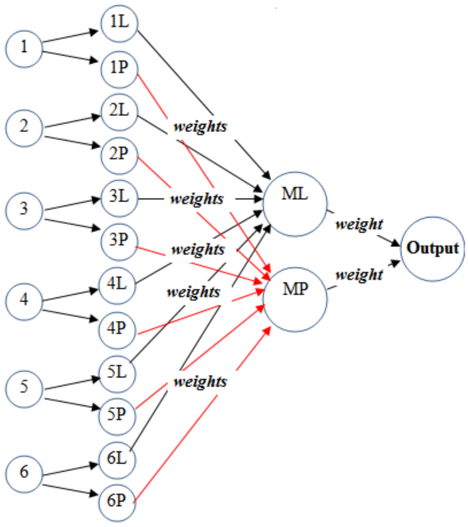

Figure 2. The neuro-fuzzy network model

The model was designed with four layers as follows:

- The first layer, called the input layer, contains six nodes that are six heuristics such as PrimaryDomain, SubDomain, PathDomain, PageRank, AlexaRank, AlexaReputation.
- The second layer contains 12 nodes. The value of each node is fuzzy value and is calculated from membership function s-shaped or z-shaped.
- The third layer contanis two nodes which are ML and MP. ML (Mean Legitimate) is the weighted sum of nodes "L" in the second layer. MP (Mean Phishing) is the weighted sum of nodes "P" in the second layer.
- The fourth layer, called the output layer, has only one the output node.

The sigmoid activation function is used in the proposed neural network, and the output value of the output node ranges from 0 to 1. The proposed model is classified into two classes so the site is phishing if the value of the output node is less than 0.5 and the site is legitimate if the value is greater than or equal to 0.5.

2) The value of six input nodes

Based on experimental results and statistics from the dataset of 11,660 phishing sites,. We found that:

- The site is a phishing site when the Levenshtein distance [17] between “PrimaryDomain”, “SubDomain”, “PathDomain” and the result of GOOGLE search engine spelling suggestion is less than 4.
- The PageRank value varies from -1 to 10. The site is a phishing site when PageRank value is low.
- The site is a phishing site when the AlexaRank value is greater than 300,000.
- The site is a phishing site when the AlexaReputation value is less 30.

Six values of the heuristics are calculated as follows:

- Calculating the value of heuristic “PrimaryDomain”: The algorithm is shown in Fig. 3.
- Calculating the value of heuristic “SubDomain” and ”PathDomain”: The algorithm is shown in Fig. 4.
- Calculating the value of heuristic “PageRank”: The Google’s PageRank value can be obtained from [18]. PageRank value varies from -1 to 10.
- Calculating the value of heuristic “AlexaRank” and “AlexaReputation”: AlexaRank and AlexaReputation value can be obtained from [19].

```
Input: PrimaryDomain
Output: Value of heuristic “PrimaryDomain”

If PrimaryDomain is IP then value= 0; // doubt phishing
If  PrimaryDomain is not IP then
        result = Suggestion_Google(PrimaryDomain);
        If result is null then
            value = 4; // no doubt phishing
        End if
        If result is not null then
            value=Levenshtein(result, PrimaryDomain);
        End if
End if
```

Figure 3. Calculating the value of the heuristic “PrimaryDomain”

```
Input: m      // m is SubDomain or PathDomain
Output: Value of heuristic “m”

If SubDomain is null then value=4; // no doubt phishing
If SubDomain is not null then
     result = Suggestion_Google(m);
     If result is null then
         value = 4; // no doubt phishing
     End if
     If result is not null then
         value= Levenshtein(result, m);
     End if
End if
```

Figure 4. Calculating the value of the heuristic “SubDomain” and ”PathDomain”

3) The value of 12 nodes in the second layer

Classifying heuristics into two linguistic labels and assigning membership functions such as s-shaped and z-shaped for each of the linguistic value. Each of these heuristic is classified into linguistic labels as “Phishing” and “Legitimate”. Equation (1) and (2) are two membership functions “s-shaped” and “z-shaped”. Based on experimental results and statistics from the dataset of 11,660 phishing sites, membership functions are calculated as follows:

$$Z(x,a,b)=\begin{cases} 1, & x\le a \\ 1-2\left(\dfrac{x-a}{b-a}\right)^2, & a<x\le\dfrac{a+b}{2} \\ 2\left(\dfrac{x-b}{b-a}\right)^2, & \dfrac{a+b}{2}<x<b \\ 0, & x\ge b \end{cases} \tag{1}$$

$$S(x,a,b)=\begin{cases}0, & x\le a\\ 2\left(\dfrac{x-a}{b-a}\right)^2, & a<x\le \dfrac{a+b}{2}\\ 1-2\left(\dfrac{x-b}{b-a}\right)^2, & \dfrac{a+b}{2}<x<b\\ 1, & x\ge b\end{cases} \qquad (2)$$

- Membership functions for "PrimaryDomain", "SubDomain" and "PathDomain": Equation (3) and (4) are two membership functions that are built to calculate fuzzy values and the graph of the membership functions is shown in Fig. 5.

$$P(x) = Z(x, 1, 3) \qquad (3)$$

$$L(x) = S(x, 2, 4) \qquad (4)$$

- Membership functions for "PageRank": Equation (5) and (6) are 2 membership functions that are built to calculate fuzzy values and the graph of the membership functions is shown in Fig. 6.

$$P(x) = Z(x, 2, 6) \qquad (5)$$

$$L(x) = S(x, 4, 8) \qquad (6)$$

- Membership functions for "AlexaRank": Equation (7) and (8) are 2 membership functions are built to calculate fuzzy values and the graph of the membership functions is shown in Fig. 7.

$$P(x) = S(x, 1mil, 3mil) \qquad (7)$$

$$L(x) = Z(x, 300k, 2mil) \qquad (8)$$

where 300k and mil are abbreviated of 300,000 and Million respectively.

- Membership functions for "AlexaReputation": Equation (9) and (10) are 2 membership functions are built to calculate fuzzy values and the graph of the membership functions is shown in Fig. 8.

$$P(x) = Z(x, 5, 20) \qquad (9)$$

$$L(x) = S(x, 10, 30) \qquad (10)$$

4) Network training algorithm

The proposed algorithm is shown in Fig. 9. The algorithm performs two phases as follows:

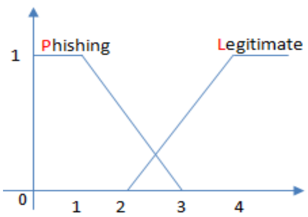

Figure 5. Graph of membership function

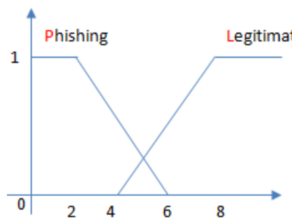

Figure 6. Graph of membership function "PageRank"

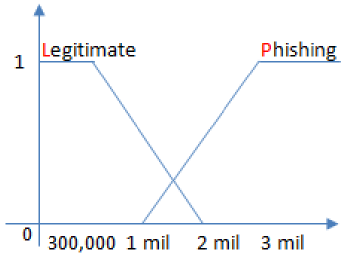

Figure 7. Graph of membership function "AlexaRank"

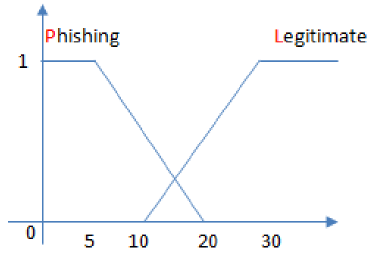

Figure 8. Graph of membership function "AlexaReputation"

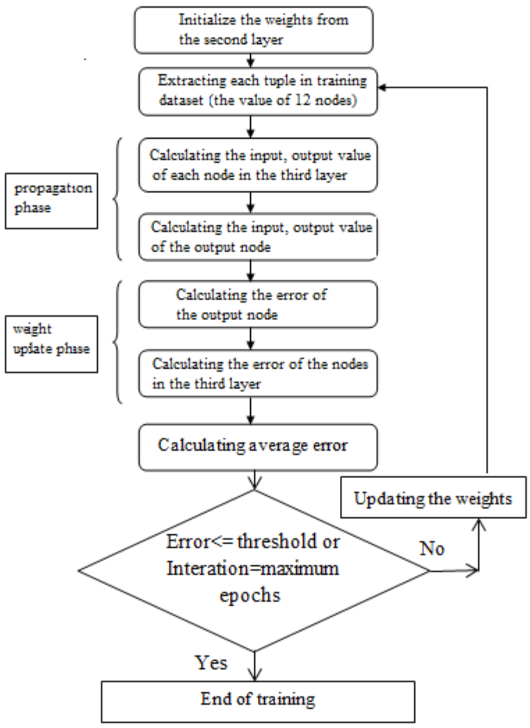

Figure 9. Network training algorithm

- The "***propagation***" phase calculates the input value, the output value of each node in the third layer and the output layer. The input value of the nodes is calculated by (11).

$$I_j = \sum_i W_{ij} O_i \qquad (11)$$

where I_j, O_i and W_{ij} are the input value of the j^{th} node in the current layer, the output value of i^{th} node in the previous layer and the weight from the i^{th} node of the previous layer to the j^{th} node of the current layer, respectively.

The output value of the nodes is calculated by (12).

$$O_j = \frac{1}{1+e^{-I_j}} \quad (12)$$

where I_j, O_j are the input value, the output value of the j^{th} node, respectively.

- The "**weight update**" phase calculates the error of the nodes in the third layer and the output layer, then updates the weights. The error of the output node is calculated by (13)

$$Err = O_o * (1 - O_o) * (T - O_o) \quad (13)$$

where T, O_O are the real value of sample in training dataset, the output value of output node, respectively.

The error of the j^{th} node in the third layer is calculated by (14)

$$Err_j = O_j * (1 - O_j) * \sum Err * W_j \quad (14)$$

where O_j, W_j and Err are the output value of the j^{th} node, the weight of the connection from the j^{th} node to the output node and the error of the output node, respectively.

The weights connect from the second layer to the third layer are updated by (15)

$$W_{ij} = W_{ij} + R * Err_j * O_i \quad (15)$$

where R, Err_j, O_i are learning rate, the error of j^{th} node in the third layer and the output value of i^{th} node in the second layer, respectively.

The weights connect from the third layer to the output layer are updated by (16)

$$W_i = W_i + R * Err * O_i \quad (16)$$

where Err, O_i are the error of output node and the output value of i^{th} node in the third layer respectively.

IV. Evaluation

We have collected 11,660 phishing sites from PhishTank [1] and 10,000 legitimate sites from DMOZ [20]. The training dataset contains 6,660 phishing sites from PhishTank and 5,000 legitimate sites from DMOZ. We built 2 testing datasets, each of which contains 5,000 phishing sites or 5,000 legitimate sites. Experimental procedure is divided into 2 phases (Training and Testing) through PHP and MYSQL.

A. Training Phase

- *Import Training Dataset*: Training dataset is imported into MYSQL. The result is shown in the Fig. 10.
- *Extracting four features of URL*: Four features (Primary Domain, SubDomain, PathDomain and Domain) are extracted. The result is shown in the Fig. 11.
- Calculating the value of six input nodes: Google search engine spelling suggestions and alexa.com are used to calculate the value of the input nodes. The result is shown in the Fig. 12.
- Calculating the value of 12 nodes in the second layer: Two membership functions s-shaped or z-shaped are used to calculate the value of the nodes in the second layer. The result is shown in the Fig. 13.
- Network Training phase: We performed the network training with 9 values of learning rate. In the training phase, the parameters are set as follows:
  - ○ Learning rate: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9
  - ○ Mean error threshold value: 1%
  - ○ Number of Epochs: 10,000
  - ○ The weights: initialize weights random values from 0 to 1

phish_id	url	phish_detail_url	submission_time	verified	verification_time
2111050	http://www.montenegrodrive.me/components /googledoc...	http://www.phishtank.com /phish_detail.php?phish_id...	2013-11-17 09:12:02	yes	2013-11-17 14:21:40
2111010	http://itunesconnect.apple.com.jooltec.com.br /upda...	http://www.phishtank.com /phish_detail.php?phish_id...	2013-11-17 09:08:17	yes	2013-11-17 13:58:52
2111001	http://kuznyanova.org.ua/deal/googledocss /googledo...	http://www.phishtank.com /phish_detail.php?phish_id...	2013-11-17 09:07:32	yes	2013-11-17 14:07:39
2110997	http://parnasseweb.tn/wp-includes /js/my.screenname...	http://www.phishtank.com /phish_detail.php?phish_id...	2013-11-17 09:07:09	yes	2013-11-17 14:08:15
2110988	http://paypal.com-inc-security-account-45453612358...	http://www.phishtank.com /phish_detail.php?phish_id...	2013-11-17 09:06:17	yes	2013-11-17 14:01:12

Figure 10. MYSQL Import

phish_id	domain	primarydomain	subdomain	pathname
2111050	montenegrodrive.me	montenegrodrive		components,googledoc,index.htm
2111010	jooltec.com.br	jooltec	itunesconnect,apple,com	updats,
2111001	kuznyanova.org.ua	kuznyanova		deal,googledocss,googledocss,sss
2110997	parnasseweb.tn	parnasseweb		wp,includes,js,my.screenname.aol.com,my.screenname...
2110988	sorpi.fr	sorpi	paypal,com,inc,security,account	cmd,home&dispatch,2f643150d63de9bd3e4d110f71b5...

Figure 11. Selecting PrimaryDomain, SubDomain, PathDomain and Domain

phish_id	primarydomain	subdomain	pathdomain	pagerank	alexarank	alexareputation
2111050	4	4	2	0	6274104	2
2111010	4	0	4	0	6274104	2
2111001	4	4	0	1	6274104	2
2110997	23	4	0	-1	160379	18
2110988	5	0	4	0	7104259	1

Figure 12. Value of six heuristics

phish_id	P1	P2	P3	P4	P5	P6	L1	L2	L3	L4	L5	L6
2111050	0.00	0.00	0.50	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
2111010	0.00	1.00	0.00	1.00	1.00	1.00	1.00	0.00	1.00	0.00	0.00	0.00
2111001	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
2110997	0.00	0.00	1.00	1.00	0.00	0.28	1.00	1.00	0.00	0.00	1.00	0.32
2110988	0.00	1.00	0.00	1.00	1.00	1.00	1.00	0.00	1.00	0.00	0.00	0.00

Figure 13. Fuzzy values

B. Testing Phase

In this phase, the proposed technique is tested with 2 testing datasets based on the weights of the network training with learning rate of 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9. RMSE (Root Mean Square Error) is a good measure of detecting accuracy. RMSE is calculated by (17).

$$RMSE = \sqrt{\frac{\sum (A_i - D_i)^2}{N}} \tag{17}$$

where D*i* is detecting sites, A_i is actual sites and N is the number of samples in testing dataset. Accuracy ratio is calculated as follows: Accuracy Ratio = 100 - RMSE. The results of the test with learning rate of 0.1, 0.2, 0.3, 0.4, 0.5, 0.5, 0.6, 0.7, 0.8 and 0.9 will be shown in Table I. From the obtained results, RMSE and accuracy are shown in Table II. We have found that the proposed technique has the best ratio of 99.10% with learning rate of 0.7 and the worst ratio of 98.22% with learning rate of 0.2 and 0.8.

TABLE I. RESULT OF TESTING WITH THE PROPOSED TECHNIQUE

Learning Rate	Testing dataset	Actual Sites (A_i)	Detecting Sites (D_i)
0.1	No.1	5,000	4,918
	No.2	5,000	4,916
0.2	No.1	5,000	4,908
	No.2	5,000	4,914
0.3	No.1	5,000	4,914
	No.2	5,000	4,931
0.4	No.1	5,000	4,939
	No.2	5,000	4,924
0.5	No.1	5,000	4,933
	No.2	5,000	4,921
0.6	No.1	5,000	4,925
	No.2	5,000	4,919
0.7	No.1	5,000	4,955
	No.2	5,000	4,955
0.8	No.1	5,000	4,914
	No.2	5,000	4,908
0.9	No.1	5,000	4,920
	No.2	5,000	4,912

TABLE II. RMSE AND ACCURACY WITH THE PROPOSED TECHNIQUE

Learning Rate	RMSE	Accuracy
0.1	1.66	98.34%
0.2	1.78	98.22%
0.3	1.56	98.45%
0.4	1.38	98.62%
0.5	1.46	98.54%
0.6	1.56	98.44%
0.7	0.90	99.10%
0.8	1.78	98.22%
0.9	1.68	98.32%

C. Comparing to Technique [10]

We experimented with the technique [10] and compared to the result of our proposed technique. First, we collect 10 testing datasets, each of which contains 1,000 phishing sites or 1,000 legitimate sites. Second, we experiment the technique [10] and the results will be shown in Table III. From the obtained result and using RMSE, we have found that the technique [10] with the accuracy of 86.06%.

TABLE III. RESULT OF TESTING WITH TECHNIQUE [10] (1):VERY PHISHY AND PHISHY (2) : VERY LEGITIMATE AND LEGITIMATE (3) : SUSPICIOUS

Testing Dataset	(1)	(2)	(3)
No.1	867	82	51
No. 2	865	76	59
No. 3	847	90	63
No. 4	902	172	26
No. 5	841	109	50
No. 6	64	873	63
No. 7	50	911	39
No. 8	39	895	66
No. 9	97	871	32
No. 10	85	863	52

D. Comparing to Technique [11]

We experimented with the technique [11] using 8 hidden nodes and hyperbolic tangent activation function. First, we collect 2 testing datasets, each of which contains 5,000 phishing sites or 5,000 legitimate sites. Second, we experiment the technique [11] and the results will be shown in Table IV. From the obtained results, RMSE and accuracy are shown in Table V, we have found that the technique [11] with the best accuracy of 94.68%.

TABLE IV. RESULT OF TESTING WITH TECHNIQUE [11]

Learning Rate	Testing dataset	Actual Sites (A_i)	Detecting Sites (D_i)
0.1	No.1	5,000	4,612
	No.2	5,000	4,520
0.2	No.1	5,000	4,624
	No.2	5,000	4,478
0.3	No.1	5,000	4,689
	No.2	5,000	4,735
0.4	No.1	5,000	4,456
	No.2	5,000	4,792
0.5	No.1	5,000	4,732
	No.2	5,000	4,736
0.6	No.1	5,000	4,721
	No.2	5,000	4,678
0.7	No.1	5,000	4,599
	No.2	5,000	4,725
0.8	No.1	5,000	4,772
	No.2	5,000	4,697
0.9	No.1	5,000	4,719
	No.2	5,000	4,699

TABLE V. RMSE AND ACCURACY WITH TECHNIQUE [11]

Learning Rate	RMSE	Accuracy
0.1	8.73	91.27%
0.2	9.10	90.90%
0.3	5.78	94.22%
0.4	8.24	91.76%
0.5	5.32	94.68%
0.6	6.03	93.97%
0.7	6.88	93.12%
0.8	5.36	94.64%
0.9	5.82	94.18%

V. CONCLUSIONS AND FUTURE WORK

We have proposed a new technique to detect phishing sites effectively. In the proposed technique, the system model is built to detect phishing sites by using neuro-fuzzy network and six heuristics (primarydomain, subdomain, pathdomain, pagerank, alexarank, alexareputation). The technique is experimented with the training dataset containing 11, 660 sites and 2 testing datasets that each dataset contains 5,000 phishing sites or 5,000 legitimate sites. The best results show that 99.10% phishing websites are detected by using the proposed technique. The proposed technique is compared to the technique [10], technique [11] and found that it is more efficient. In the future, the proposed neuro-fuzzy model will be improved to enhance the detection ratio. Besides, the system could be furthermore enhanced by using larger datasets and more heuristic parameters.

REFERENCES

[1] PhishTank. (Nov. 2013). Statistics about phishing activity and phishtank usage. [Online]. Available: http://www.phishtank.com/stats/2013/01/

[2] D. Goodin. (2012). Google bots detect 9,500 new malicious websites every day. [Online]. Available: http://arstechnica.com/security/2012/06/

[3] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang. (2009). An empirical analysis of phishing blacklists. [Online]. Available: http://ceas.cc/2009/papers/ceas2009-paper-32.pdf

[4] McAfee. (July 2011). Mcafee site advisor. [Online]. Available: http://www.siteadvisor.com

[5] Y. Zhang, J. I. Hong, and L. F. Cranor, “Cantina: A content-based approach to detecting phishing web sites,” in *Proc. 16th International Conference on World Wide Web*, 2007, pp. 639–648.

[6] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, “Cantina+: A feature-rich machine learning framework for detecting phishing web sites,” *ACM Transactions on Information and System Security*, vol. 14, no. 2 , pp. 1–28, Sept. 2011.

[7] M. E. Maurer and D. Herzner, “Using visual website similarity for phishing detection and reporting,” in *CHI ’12 Extended Abstracts on Human Factors in Computing Systems*, 2012, pp. 1625–1630.

[8] A. Sunil and A. Sardana, “A pagerank based detection technique for phishing web sites,” in *Proc. IEEE Symposium on Computers & Informatics*, 2012, pp. 58–63.

[9] M. G. Alkhozae and O. A. Batarfi, “Phishing websites detected based on phishing characteristic in the webpage source code,” *International Journal of Information and Communication Technology Research*, vol. 1, no. 6, pp. 283–291, Oct. 2011.

[10] M. Aburrous, M. Hossain, F. Thabatah, and K. Dahal, “Intelligent phishing website detection system using fuzzy techniques,” in *Proc. Third International Conference on Information and Communication Technologies: From Theory to Applications*, 2008, pp. 1–6.

[11] N. Zhang and Y. Yuan. Phishing detection using neural network. CS229 lecture notes. [Online]. Available: http://cs229.stanford.edu/proj2012/ZhangYuan-PhishingDetectionUsingNeuralNetwork.pdf, 2012

[12] Y. Norazah, B. A. Nor, S. O. Mohd, and C. N. Yeap, “A concise fuzzy rule base to reason student performance based on roughfuzzy approach,” in *Fuzzy Inference System-Theory and Application*, InTech, 2010.

[13] R. Ata and Y. Kocyigit, “An adaptive neuro-fuzzy inference system approach for prediction of tip speed ratio in wind turbines,” *Expert Systems with Applications*, vol. 37, no. 7, pp. 5454–5460, 2010.

[14] J. S. R. Jang, “ANFIS: Adaptive-network-based fuzzy inference system,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[15] K. S. Pal and S. Mitra, *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing*, New York: JohnWiley & Sons, 1999.

[16] Wikipedia. (2014). [Online]. Available: http://en.wikipedia.org/wiki/Uniformresourcelocator

[17] Levenshtein. (2014). [Online]. Available: http://en.wikipedia.org/wiki/Levenshtein_distance

[18] G. Inc. (2014). [Online]. Available: http://toolbarqueries.google.com

[19] Alexa. (2014). [Online]. Available: http://data.alexa.com/data?cli=10&dat=snbamz&url=

[20] DMOZ. [Online]. Available: (2014) : http://rdf.dmoz.org/rdf/

Luong Anh Tuan Nguyen is a lecturer in the Faculty of Information Technology, HoChiMinh City University of Transport, Vietnam. He received B.Sc and M.Sc in Computer Science from HCM City University of Science. His current research interests include intelligent control, fuzzy systems, neural network, security and cloud computing. Currently, he is also Ph.D student.

Ba Lam To is a lecturer in the Faculty of Information Technology, HoChiMinh City University of Transport, Vietnam. He received his B.Eng. degree in Information Technology from HCM City University of Technology in 2006 and received Master and Ph.D degree in 2008 and 2012 respectively from Université Pierre et Marie Curie (UPMC - Paris 6). His current research interests include Cloud Computing, Routing in Cognitive Radio Network, Network Security, Intelligent transportation system. Currently, he is vice dean of IT Faculty. He also serve as a research director of the Faculty.

Huu Khuong Nguyen received Ph.D degree in automation control engineering from the Russian Academy of Sciences in 1999. Since 2005, he has been an associate professor. He chaired a lot of research projects. Currently, he is vice president of HoChiMinh City University of Transport, Vietnam. He is responsible for scientific research.

