Intelligent phishing detection parameter framework for E-banking transactions based on Neuro-fuzzy

P. A. Barraclough
Computer Science and Digital Technology
Univeristy of Northumbria
Newcastle Upon Tyne, NE 18ST, United Kingdom
Email:phoebe.barraclough@northumbria.ac.uk

G. Sexton
Computer Science and Digital Technology
Univeristy of Northumbria
Newcastle Upon Tyne, NE 18ST, United Kingdom
Email:g.sexton@northumbria.ac.uk

M.A. Hossain
Computer Science and Digital Tecnology
University of Northumbria
Newcastle Upon Tyne, NE1 8ST, United Kingdom
Email:alamgir.hossain@northumbria.ac.uk

N. Aslam
Computer Science and Digital Tecnology
University of Northumbria
Newcastle Upon Tyne, NE1 8ST, United Kingdom
Email:nauman.aslam@northumbria.ac.uk

Abstract— **Phishing attacks have become more sophisticated in web-based transactions. As a result, various solutions have been developed to tackle the problem. Such solutions including feature-based and blacklist-based approaches applying machine learning algorithms. However there is still a lack of accuracy and real-time solution. Most machine learning algorithms are parameter driven, but the parameters are difficult to tune to a desirable output. In line with Jiang and Ma's findings, this study presents a parameter tuning framework, using Neuron-fuzzy system with comprehensive features in order to maximize systems performance. The neuron-fuzzy system was chosen because it has ability to generate fuzzy rules by given features and to learn new features. Extensive experiments was conducted, using different feature-sets, two cross-validation methods, a hybrid method and different parameters and achieved 98.4% accuracy. Our results demonstrated a high performance compared to other results in the field. As a contribution, we introduced a novel parameter tuning framework based on a neuron-fuzzy with six feature-sets and identified different numbers of membership functions different number of epochs, different sizes of feature-sets on a single platform. Parameter tuning based on neuron-fuzzy system with comprehensive features can enhance system performance in real-time. The outcome will provide guidance to the researchers who are using similar techniques in the field. It will decrease difficulties and increase confidence in the process of tuning parameters on a given problem.**

Keywords*—*FIS, Intelligent phishing detection, fuzzy inference system, neuro-fuzzy

I. INTRODUCTION

Phishing is a fraudulent mechanism using both social engineering and technical deception to obtain user's sensitive information and financial account credential for financial benefit. Phishing techniques have become a major concerned in web-based transactions causing monitory losses annually. According to the Accords Association's Press report, an increase in phishing attacks in online transaction caused losses of £21.6 million between January and June 2012, which was a growth of 28% from June 2011[1]. Due to this problem, various anti-phishing approaches have been proposed to solve the problem. These approaches include feature-based applying machine learning techniques [2], [3], blacklist-based approaches, using machine learning techniques [4], [5], [6], [7], and content-based applying machine learning algorithms have also attempted to solve the problem [8], [2]. However, there are still high false positive causing inaccuracy and a lack of real-time solution. These machine learning techniques require parameter settings. However parameters are difficult to set to a desirable output and there is a lack of parameter tuning framework [9], particularly for phishing website detection.

Generally, phishing detections are divided into two main categories: Phishing emails level and phishing websites level. This study focuses on phishing website detection on feature-based including content-based, using machine learning techniques. The most common machine learning algorithms including logistic regression, fuzzy logic, neural network, perceptron and many more

The main phishing website detection approaches are either utilizing: (1) Feature-based including content based approaches applying machine learning algorithms to discriminate between legitimate sites and illegitimate sites or (2) URL blacklist-based approach that uses a list of URL of known illegitimate websites.

As the main contribution, the study has introduced parameter tuning framework based on a neuro-fuzzy algorithm, using 6 feature-sets by identifying different numbers of membership functions, different numbers of epochs and different numbers of feature-sets. This is a novel work that has not been considered in the literature in a single platform in this field.

The question is: How can parameter tuning framework based on neuro-fuzzy system with 6 feature sets be used to enhance phishing detection system performance in real-time?

The aim is to identify features from diverse sources and develop parameter tuning framework based on neuro-fuzzy system with six sets of inputs that can be used by researchers in the field. Specific objectives are: (1) to identify

comprehensive features for training and checking/testing, (2) to develop fuzzy models for parameter tuning framework from given features, based on Neuron-fuzzy, using different feature-sets, (3) to train and check/test the models using cross-validation methods, and (4) to conduct a comparative study to prove the capability and merit of the parameter tuning framework.

The outcome is expected to provide guidance to the researchers who are using similar techniques. It will decrease difficulties and increase confidence in the process of tuning parameters on a given problem.

To gather features, 84% features were taken from our previous work, 3.5% were extracted from phishing websites from phish Tank archive, legitimate site rules. Features were explored using European Union Agency for Network and Information Security (ENISA) website, [10] and User's credential profile. The remaining 12.23% were identified from Journals. This research was done while based at the school of Engineering and Environment at Northumbria University. The units of analysis are 6 individual frameworks.

The remaining sections are as follows: Section II covers literature review. Section III describes methodology including feature gathering and Analysis. Section IV covers experimental set up. Section V covers experiments including Framework, training and testing. Section VI presents results and discussions. Analysis is presented in section VII. Section VIII concludes the paper and provides future work.

II. RELATED WORK

Phishing attacks have increased and are becoming sophisticated, which have led to $15 billion losses in the global economy in 2012 [1]. This has caused a number of phishing solutions to be developed to tackle the problem. Anti-phishing detection solutions mainly utilize two approaches: feature-base approaches that utilize Uniform Resource Locator (URL), blacklist-based and approaches that utilize features-based including content, using machine learning techniques.

A. Content-based through Machine Learning techniques

Major researches have considered content-based approaches based on machine learning techniques to detect phishing websites [2], [11], [12], [13], [14], [15], [16].

Aburrous proposed a model to identify electronic banking sites [2]. The method utilized a combined fuzzy logic and data mining algorithms, using twenty seven characters and factors that identify phishing websites. Their approach achieved 84.4%, but suffered 15.6% error rates, which is a high risk for online users.

In an attempt to improve the detection approaches, Suriya proposed fuzzy logic, using factors and a case study to assess whether phishing attack was taking place or not [11]. Their method employed three layered checker in web pages to check for tricks of attackers, using JavaScript to hide data from users. The result revealed that their approach can detect phishing 96% correctly. However using only 3 layer method to detect phishing is limited since phishing techniques are varied.

Similarly, Wenyin considered a method based on reasoning of Semantic Link Network, using 1000 illegitimate web pages and 1000 legitimate web pages to directly discover the target name if it is a phishing website or a legitimate website [12]. Their approach had ability to identify phishing sites using inferring rules. Wenyin, however, acknowledged that the model suffered 16.6% false negative and 13.8% false positive, which are high level of error rates.

Equally, Xiang explored content-based probabilistic method that incorporates URL blacklists with shingling algorithms utilized by search engine and information retrieval technologies (IRT) to identify phishing websites [13]. Their approach had advantage of using TF-IDF and a scoring function in the search engine, when they match queries to pages that produces a probabilistic framework for detecting phishing sites. The experimental result was 67.74% and 73.53% accuracy with 0.03% error rates. Although this method has low false positives, its accuracy can make user vulnerable to phishing attacks.

Moreover, Dong focused on defending the weakest link in phishing websites detection, by analyzing online user behaviours based on visited websites and the data a user submitted to those websites [14]. Taking user's behavior into consideration is important in addressing phishing attack, but only dealing with the data users submitted to detect phishing sites is a major limitation in handling a well designed phishing websites.

Likewise, Wardman came along with a new method using file matching algorithms, hashing function index MD5 hash value and Deep MD5 Matching, to decide if a file can be utilized to classify a new file in the same group of phishing web pages [15]. Their method was tested to identify the system performance. The results demonstrated that their technique could achieve more than 90% in performance. However, the approach suffered high level of false positive rates (10%).

In the attempt to improve phishing detection scheme, Barraclough proposed a novel method to detect phishing website [16]. The approach was based on machine Neuro-fuzzy, using five sets of inputs with 288 features, which offered accuracy results of 98.4%. This result demonstrated high accuracy, but suffered 1.6% error rates. Their finding was that a hybrid neuro-fuzzy with 5 input feature-sets can detect phishing websites with high accuracy in real-time.

B. URL Blacklis-based Approaches

Another study explored blacklist-based that uses a list of URL of known illegitimate websites [4], [5], [7], [17], [18], [19], [20], [21]. For instance, Xiang proposed blacklist and content-based model to strengthen human-verified blacklist by using probabilistic techniques to obtain higher accuracy [4].

Their experiment obtained 87.42% true positive, but suffered 4.34% false positives, which is a high error rates.

Similarly, Ma conducted a study and explored phishing website detection [5]. Their approach was based on machine learning algorithms consisting of Support Vector Machine (SVM), Logistic Regression (LR) and Naïve Bayes (NB), using 10,000 host-based features from WHOIS queries with Lexical features to classify website reputation on the relationship between the lexical and host-based features. Their approach yielded 95% and 99% accuracy, and error rates range of 0.9% and 3.5%. However, Ma acknowledged that their method could not handle large evolving phishing websites that are created regularly [5].

Equally, Whittaker designed Google's phishing classifier to automate the maintenance of Google's blacklist [7]. Their method was based on logistic regression classifier, using URL-based lexical features, web page content and Hypertext Markup Language (HTML) to automatically classify phishing web pages. Their experimental results achieved 90% accuracy in real-time with 10% error rates. However, Whittaker recognized that their blacklist keeps behind with update and can only identify phishing site after it has been published and appeared on the Internet [7].

Similarly, PhishDef was developed by Le [17]. Their method was based on URLs lexical features, using algorithms to compare phishing websites. Their features were evaluated utilizing online learning algorithms including batch-based Support Vector Machine (SVM), Online Perceptron (OP), Confidence Weighted (CW) and Adaptive Regularization of Weights (AROW) that overcomes noisy data when detecting phishing websites. For each URL inputs, the classifier makes a decision whether a website is suspicious or not. Their approach achieved an average of 97% accuracy using offline algorithms and 90% using online algorithms. However, Le's research suffered features inadequacy, which is a similar problem to the study of Xiang [4]. Le's study is related to the study of Ma in their methodology. Both methods used URL feature-based [17], [5].

In addition, Huh and Kim applied search engines to measure URL which identified phishing websites and ranked them below 10, while legitimate sites were ranked top [18]. For evaluation performance, Google, Bing and Yahoo were used. As well as this, 100 legitimate websites and 100 illegitimate websites were employed, applying classification algorithms to measure website reputation including linear discrimination analysis, Naïve Bayesian, K-Nearest Neighbour and Support Vector Machine. Using K-Nearest Neighbour achieved accuracy of 95% and 6.2% error rates. Although K-Nearest Neighbour performed better in comparison with the best classifiers, URL features alone is limited to detect phishing websites, while legitimate websites can be compromised easily by attackers and spoil their validity.

Canali proposed Prophiler, a lightweight malware static filter, using HTML, JavaScript and URL with features through a classifier that identifies non-malicious pages to assess more malicious pages to a great extent [19]. While Prophiler was intended to be a fast filter, it allows higher false positive rates in order to reduce false negative rate. In addition, CANTINA+ was proposed by Xiang [20]. The approach was based on machine learning techniques, using URL, Search Engines, the HTML Document Object Model (DOM) and PhishTank with fifteen features. Although the results revealed 92% accuracy, it suffered 8% error rates. Furthermore, Ead proposed a combination of artificial immune systems and Fuzzy systems with both lexical and host-based URL features [21]. The advantage of this approach is that it classifies URLs automatically as phishing or legitimate sites.

Although the above mentioned approaches are effective to some degree of accuracy, there are still high false positive rates due to a lack of adequate features and a lack of proper parameter tuning [9], [22]. Therefore, this study addresses the problem by introducing a novel parameter tuning framework with comprehensive features based on neuro-fuzzy system to maximize phishing detection system performance.

III. METHODOLOGY

Despite a number of existing state-of-the-art feature-based, using machine learning techniques to detect phishing attack, there are still high false positives. The study of Jiang, Ma and Xiang found that this problem is caused due to a lack of comprehensive features and a lack of parameter tuning framework [9], [22], [23]. Based on the findings by Ma and Jiang, the aim is of this study is to introduce parameter tuning framework based on neuro-fuzzy, using comprehensive features to detect phishing website.

Neuro-fuzzy is a combination of fuzzy logic and neural network. It is a network structural consisting of nodes and connections through which nodes are linked. Parts of nodes are adaptive meaning that their outputs relies on applicable parameters, learning rules identifies how these parameters can be set to reduce error measures [9]. The choice of Neuro-fuzzy is that it has the advantage of both neural network which is capable of learning new data and fuzzy logic which deals with linguistic values as well as making decisions using fuzzy [If-Then] rules [9]. Neuro-fuzzy also creates input-output map, which is most practical for the set objectives [24]. Our methodology process is illustrated in Fig. 13.

To tackle this problem robustly, Fig. 13 shows that first, we identify a diverse spectrum of sources to extract feature that characterize phishing techniques, using the knowledge provided in a relevant journals [16], legitimate site rules including European Union Agency for Network and Information Security (ENISA), website [10], 'Complying with anti-phishing regulation' website, [25], phishTank archive [26], and user-behavior profile [1]. Secondly, we split features into different size of sets and split each set into pairs. Parameters are defined for each framework, initial structure is generated, training and testing is performed and outputs are view. Feature gathering and analysis are discussed further in section A.

A. Feature gathering and Analysis

We used qualitative features based on quantitative results. Features are also used interchangeable with data. Overall 342 features were utilized for training and testing. Specifically, while 288 features were acquired from our previous work [16], 12 features shown on Table I are newly introduced and were extracted from secondary sources by exploring the existing knowledge as follows: Based on legitimate site rules ENISA's website was used and 'Complying with anti-phishing regulation' website to extract feature including: (1) Data Protection Act designed to protect personal data, (2) *Bill C-28, CAN SPAM* is a new Canada bill of 2010 requires that all email senders to obtain prior consent from recipient, (3) *Web copyright*: one of the tactic used by phishing is to create a website that looks real, but in reality it is illegitimate website, and (4) *Phishing criminal* are phishing gangs. This list has weight value 0.1- 0.5. The remaining 54 features are from the works by Abu-Nimeh and Xiang [27], [22].

Similarly, based on phishTank archive, we explored knowledge given from phishing websites to extract features including: (1) Visual deception: phishing visual deception technique is intended to imitate legitimate site images and text, (2) Perception reality: perception in visual environment don't always match the reality, if phishers creates images and logo perfect as exact copies, (3) Neglect warning: users are more likely to ignore bad design of user warning interface provided to alert users, and (4) Legitimate websites: are supposed to be exactly as their purpose. This list has weight value range of 0.1 - 0.5.

Equally, based on user-behaviour profile, we explored knowledge from relevant journals to extract features as follows: (1) Redirect to illegitimate website: the actual URL is usually directed to a different website that was not intended by the user, (2) Web page content: have text where the letter 'I' is substituted by the letter 'L' or the number '1' to fool users about the true domain names, (3) Other items: phishing also take the advantage of user's lack of security knowledge as well as lack of attention to security indicators, and (4) Web browser document: another phishing trick is to put a phishing browser document on top of a legitimate window to trick users. This list has weight value range of 0.1 - 0.5. These 12 features were gathered during the period between 5th and 10th July 2013. Features from phishTank were extracted using an automated wizard and all features were stored in Excel worksheet because it offers a format ready for MATLAB.

Most frequent terms was performed across features using the 'find' function to identify features. The features were prepared using normalization method by assigning weight to each feature using a value range between [0 and 1]. While 0 (zero) is low, 1 (one) is high and there is an in between numbers. This normalization is done in order to remove effects that occurs in features to make sure that the impact of technical bias are reduced in the results. Table I shows that features are divided into groups of 3 rows, and the first 4 features are assigned a weight of 0.5 which indicates that the features with 0.5 weight have high importance in combating phishing, while the features with 0.3 weight are moderate, 0.1 indicates low. The advantage is that features are used to generate models and fuzzy If-Then rules. Moreover, we use features to discriminate between phishing, suspicious and legitimate sites accurately and in real-time.

B. Feature size

The choice of a total of 342 features size is adequate to produce a desirable output (Huange et al., 2006).

C. Limitation

The challenge in using neuro-fuzzy is that input membership function parameter is limited to either constant of linear.

IV. EXPERIMENTAL SET-UP

For our experiment MATLAB fuzzy logic tool box was used because it has FIS editor and other four integrated editors which are useful for our training and testing process. Cross validation methods are used to validate the model. A number of Cross validation methods exist, such as 20-Fold CV, 10-Fold CV, 5-Fold CV, 2-Fold CV and LOOCV, but 2-Fold CV and 10-fold CV were used in this paper because they can handle the conventional data well [28]. Before training, the data-sets are split into training pair and testing pair. The training pair was used to generate fuzzy models and to train the model, while test pair was used for testing the models and to check its merits. Checking, also handles the model overfitting during the training process [28].

A. Parameter Framework Descriptions

Parameter tuning framework for intelligent phishing detection, using a Neuro-fuzzy as presented in Table II. It specifically shows parameter optimal specification that has impact in fuzzy system performances. Column 1 shows that 6 experiments were run for every framework. Numbers of input membership functions (MFs) were assigned to each individual run. Column 3 demonstrates that output membership functions are linear. Column 4 presents numbers of parameters, while varieties of epochs were given in column 5 which presents the number of iterations. Column 6 gives a range of training data-sizes for each run and checking data-sizes are provided in columns 7 as illustrated on Table II. The results and analysis of these experiments will be presented in section 5 and 6. The best performance will be highlighted.

TABLE I. FEATURES THAT CHARACTERIZE PHISHING WEBSITE

No	Features	Layers
1	Visual deception	0.5
2	Data Protection Act	
3	Redirect to illegitimate website	
4	Perception reality	
1	Phishing criminals	0.3
2	BILL C-28, CAN SPAM	
3	Webpage contents	
4	Neglect warning	
1	Webpage copyrighted	0.1
2	Legitimate website	
3	Other items	
4	Web browser document	

TABLE II. PARAMETER TUNING FRAMEWORK SPECIFICATION

Experiment	Input MF No	Output MFs	Number of Parameter	Epochs Number	Training set	Checking/test set
Framework1	5	Linear	30	10	150	150
Framework2	5	Linear	15	10	151	151
Framework3	3	Linear	15	12	171	171
Framework4	3	Linear	15	10	114	114
Framework5	3	Linear	15	9	57	57
Framework6	3	Linear	30	30	28	28

V. EXPERIMENTS AND ANALYSIS

The aim of this paper is to design parameter tuning framework for phishing detection utilizing Neuro-Fuzzy system. Practically, rules are determined by expert in expert systems. In supervised learning, algorithms are trained on inputs where the desired outputs are known. Thus, all input and output membership function parameters assigned are selected empirically by determining the desired input and output. Since there is no easy way to decide the smallest number of the hidden nodes essential to obtain a preferred level of performance, adjustments are done after evaluation. Section A begins by identifying the specifications required for parameter tuning framework experiment, section B presents framework1, section C present phishing detection Fuzzy Inference System (FIS) process. Section D discusses testing process. Section E presents Framework2, section F presents framework3, section G presents framework4, section H presents framework5, section I presents framework6.

A. Training

To perform training and testing for the parameter tuning framework, first features were randomly split into series of training and testing sets that carries the desired inputs to outputs. A Cross validation (CV) methods was also applied to train and test the parameter tuning framework models for reliability using 2-Fold CV on framework 1, 2 and 6, while using 10-fold CV method on frameworks 3, 4 and 5. Both cross-validations were used since they can handle conventional features well given the 342 feature-set [28]. Training sets are used to generate fuzzy models, fuzzy rules and to train the model. Testing sets are used to check the generalization capability of the fuzzy models and to handle over-fitting that occurred during training process. Individual framework was assigned different sizes of training and testing sets, different numbers of input and output membership functions, parameter optimization methods, different numbers of epochs and different numbers of error tolerance.

B. Framework1

- A total of 300 features were utilized in Framework1, which are split into 150 training set and 150 test set. The training set is utilized to generate a model and to train the fuzzy model while the remaining 150 set is utilized for testing the model.
- 5 membership functions are assigned for the input.
- Linear is set for the output membership functions.
- Parameter optimization methods are assigned to hybrid, back-propagation and least square
- 10 epochs are assigned so that after 10 iterations, the process stops at the minimal error tolerance which is assigned to zero tolerance.

C. Phishing Detection fuzzy Inference System (FIS) Structure

A model similar to Sugeno type was generated and presented in Fig. 1. The structure consists of five functional components: Input Layer, Fuzzification, Rule base, Normalisation, and defuzzification [9]. The processes of fuzzy reasoning performed by FIS based on rules include:

a) Layer 1: This is the input layer. Neuron in this step simply transmits crisp straight to the next layer.

b) Layer 2: is fuzzification. In this layer, inputs are taken and classified into a degree of membership functions in which they belong to as fuzzy sets. This is shown in Fig. 2.

c) Layer 3: is a Rule base where all the rules are assigned weight between [0 and 1]. For every rule, implication is implemented that generates qualified consequent as a fuzzy set of each rule depending on the firing strength. A rules-base sample containing 5 fuzzy IF-THEN rules generated through framework1 experiments is presented in Fig. 6.

d) Layer 4: is Aggregation. In this layer, each rule is combined to make a decision. The output of the aggregation process is a fuzzy set whose membership function assigns a weighting for each output value.

e) Layer 5: is defuzzification. In this layer, the input for the defuzzification process is acombined output fuzzy set and the output is a single number. The most common defuzzify method is the centroid calculation [9].

Fig. 1, a FIS model shows that given the values of premise parameters, the overall output is expressed as *linear* combining consequent parameters. Hybrid learning algorithm is used as parameter optimization method to enhance performance. In the forward pass for that particular algorithm, functional signals move forward until layer 4. Then consequent parameters are classified by the least square estimate (LSE). The error rates in the backward pass get propagated backward, while the premise parameters get updated using the gradient descent [9].

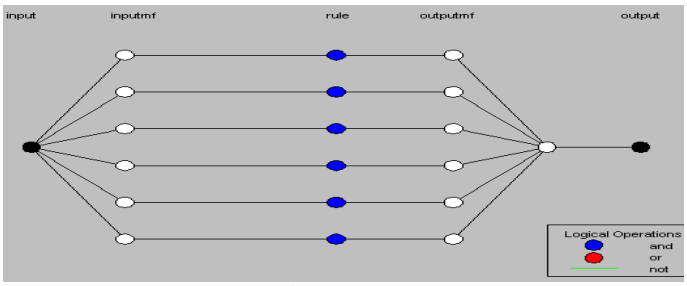

Fig 1.(a) , Framework1: Fuzzy inference model

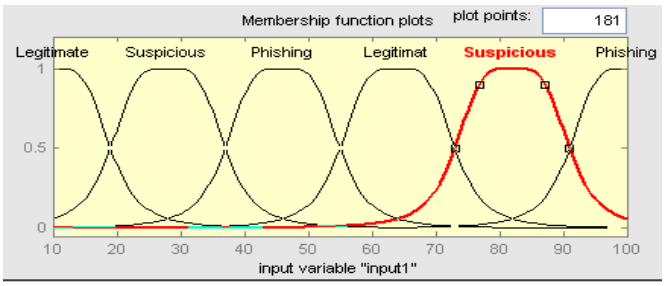

Fig. 1 (b), Framework1: 5 membership functions after training.

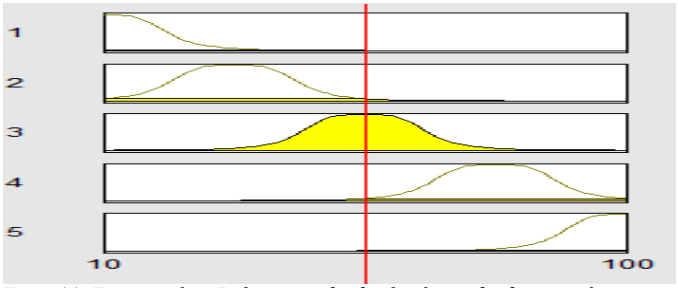

Fig. 2 (a), Framework2: Rule viewer for final risk rate for framework2

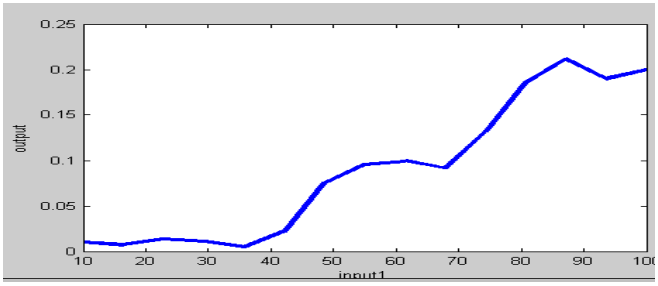

Fig. 2 (b), Framework2: Performance evaluation graph for framework2

D. Testing

After the training was completed, the checking set was used to check and to test the model. The training and testing process was repeated two times the fold, utilizing training and checking sets only once. This process was repeated 6 times for each individual experiment as shown in Table II

The results were observed. Training outputs are presented in Fig. 1 (b), while test results are presented in section V1 as shown in Table III, columns 2, 3 and 4. Fig. 1, (b) is the output for input membership function, type Gbell membership function with the value range of [0, 1] in Y-axis and a value range between [10, 100] on the X-axis. It is defined by linguistic terms: low as legitimate, medium as suspicious, while high as phishing.

E. Framework2:

- In Framework2, a total of 302 features were utilized. These are split in training pair to generate fuzzy model and testing set.
- The first 151 pair is used for training the model while the remaining 151 pair is used to validate the identified model
- 5 membership functions were assigned for the input.
- Linear is set for the output membership functions.
- Parameter optimization method are assigned to hybrid, back-propagation and least square
- 10 epochs are assigned so that after 10 iterations, the process stops at the minimal error tolerance which is assigned to zero tolerance. The training outputs are presented in Fig. 2 (a) and Fig. 2 (b).

F. Framework3

- In Framework3, a total of 342 features were utilized, which were split in training set and testing set. Training set was used to generate fuzzy model and rules.
- The first 171 set was utilized create a fuzzy model and to train the model, while the remaining 171 set was utilized to validate the model.
- 3 numbers of input membership functions were assigned.
- Linear method was assigned for the output membership functions.
- Parameter optimization method were assigned to hybrid, which is back-propagation and least square
- 12 epochs a assigned so that after 12 iterations, the process stops at zero error tolerance. The training results were observed and presented. The training output is shown in Fig. 3 (b).

G. Framework4

- In Framework4, a total of 228 features are utilized, which are split into 114 training set and 114 test set. 114 training set was utilized to generate a model and to train the fuzzy model while the remaining 114 set was utilized to test the model.
- 3 numbers of input membership functions were assigned.
- Linear was set for the output membership functions.
- Parameter optimization, hybrid method was assigned, a back-propagation and least square.
- 10 epochs was assigned so that after 10 iterations, the process stops at zero minimal error tolerance.

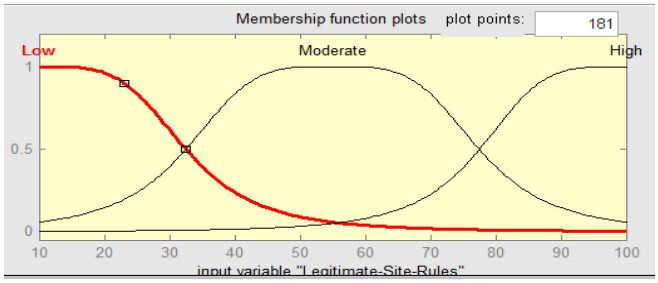

Fig. 3(a) Framework3: Membership Functions (MFs) after training

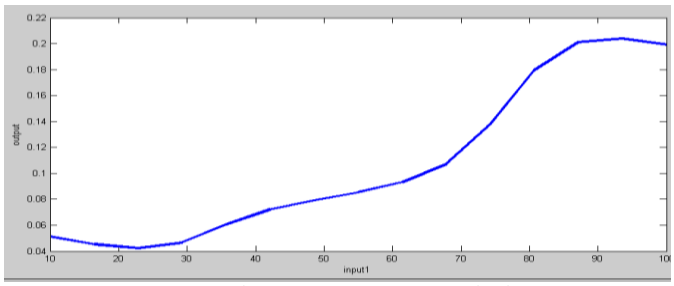

Fig. 3 (b) Framework3: Performance evaluation graph for framework3

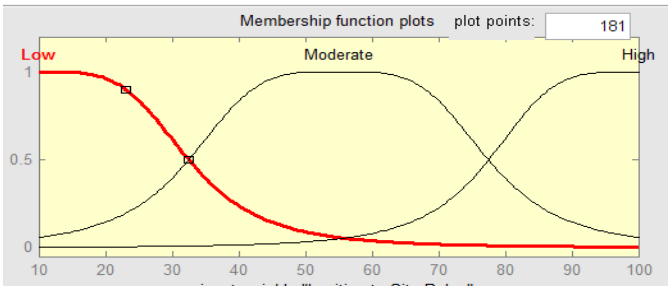

Fig. 4 (a) Framework4: Membership Functions after training

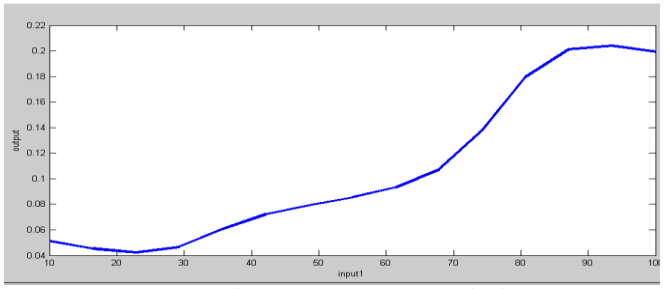

Fig. 4(b) Framework4: Performance evaluation graph for framework4

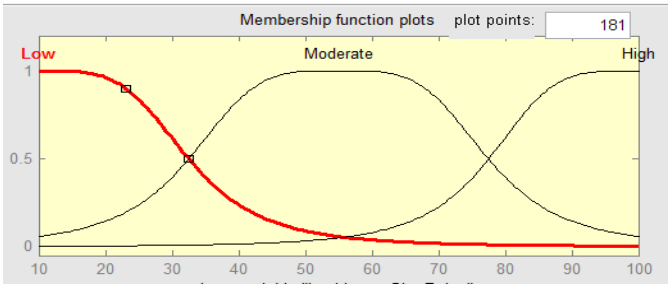

Fig. 5. (a) Framework5: Membership Functions (MFs) after training

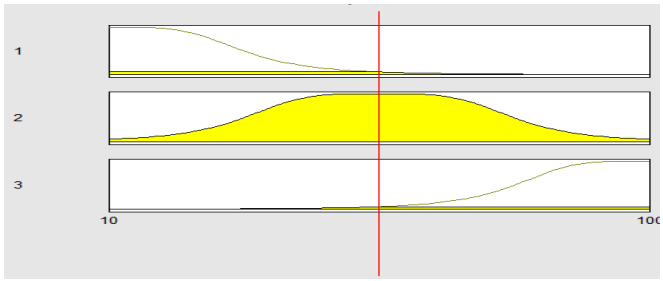

Fig. 5. (b) Framework5: Rule viewer for final risk rate for framework5

.

The training results are presented and observed. The framework4 training outputs are similar to Framework3 training outputs.

H. Framework5

- In Framework5, a total of 114 features were utilized, which are split into 57 training set and 57 test set. 57 training set was utilized to generate a model and to train the fuzzy model while the remaining 57 set was utilized to test the model.
- 3 numbers of input membership functions were assigned.
- Linear was set for the output membership functions.
- Parameter optimization, hybrid method was assigned, a back-propagation and least square
- 10 epochs was assigned so that after 10 iterations, the process stops at zero minimal error tolerance. The training results are presented. The training outputs are presented in Fig. 5 (a) and Fig. 5 (b).

I. Framework6

- A total of 56 features were utilized in Framework6, which are split into 28 training set and 28 test set. 28 training set was utilized to generate a model and to train the fuzzy model while the remaining 28 set was utilized to test the model.
- 3 numbers of input membership functions were assigned.
- Linear was set for the output membership functions.
- Parameter optimization, hybrid method was assigned, a back-propagation and least square
- 30 epochs was assigned so that after 30 iterations, the process stops at zero minimal error tolerance. All testing results are presented in section V1.

J. Basic Rules

Fuzzy IF-THEN rules are expressed in the form:
If A Then B, where A and B are labels of fuzzy sets [29] characterized by appropriate membership functions. Regarding their concise form, fuzzy if-then rules are usually utilized to obtain the imprecise modes of reasoning that does an important role in the human ability to decide in an environment of uncertainty and imprecision. A description of a simple fact in phishing detection is: If the risk is high or 100% risk, then it is a phishing. If the risk is 0% risk then it is a legitimate. Any number of risks between 0% to 100 is suspicious. An example of rules is shown in Fig. 6 for framework1. It is different for each framework depending on

the number of MFs. If-Then rules are used because fuzzy rules have been widely utilized successfully in controls and modeling [15].

Table III. TESTING RESULTS USING DIFFERENT DATASET-SET SIZES AND A SELECTION OF PARAMETERS FOR TUNING

Experiments	Average test Error	Average test Error %	Test Accuracy	Parameter Number	Training Set	Checking Set
Framework1	0.017018	1.7%	98.3%	30	150	150
Framework2	0.016961	1.7%	98.3%	15	151	151
Framework3	0.016283	1.6%	98.4%	15	171	171
Framework4	0.016283	1.6%	98.4%	15	114	114
Framework5	0.16297	1.6%	98.4%	15	57	57
Framework6	0.017147	1.7%	98.3%	30	28	28

If input1 is Legitimate then output is out1mf1 = 1
If input1 is Suspicious then output is out1 mf2 =1
If input1 is Phishing then output is out1 mf3 =1
If input1 is Legitimate then output is out1 mf4 =1
If input1 is Suspicious then output is out1 mf5 = 1

Fig. 6. Rule base containing 5 fuzzy IF-THEN rules

VI. TESTING RESULTS AND DISCUSSIONS

After conducting extensive experiments, results are obtained on average error which is a measure of the model accuracy performance. The exact measurement is the overall output in which the model will be compared. In this section, 6 model results are presented on Table III and Figs. 7, 8, 9, 10, 11, and 12. Blue crosses on graphs indicate training results, while red stars indicate test results. Overall detailed results for all 6 models are shown on Table III. Average testing errors in column 3 are rounded to 2 decimal places and converted to percentage obtain the accuracy as shown in Table III

VII. ANALYSIS

The parameter tuning framework was evaluated using 2-fold and 10-fold cross-validation methods. Framework3 and Framework4 were assigned a number of 15 parameters. 3 and 4 MFs were also specified, using 12 and 10 Epochs. 0.016283 average errors were obtained, which demonstrated best results compared to the other 4 models. Framework5 followed by achieving 0.016297 average errors in which the difference can only be seen on a fine scale. This was evaluated on 2-fold cross-validation, 15 numbers of parameters, assigned 3 numbers of membership functions and 9 epochs. Model 6 suffered on average error of 0.017147 with a difference of 0.1% compared to others. The lower the average error rates, the better the results. The highest result achieved is nearer to the expected results, given the target performance to be closer to 100% accurate if not 100% accurate. In which case, 98.4% accuracy is nearer enough.

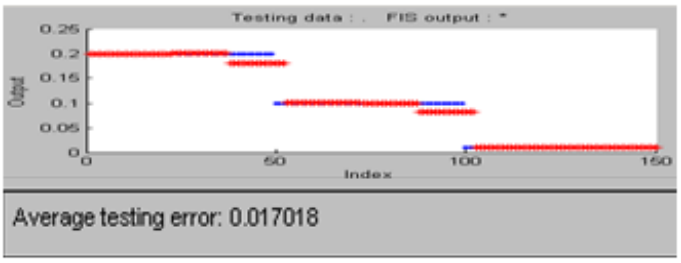

Fig. 7 Result for Framework1 with 5 input MFs

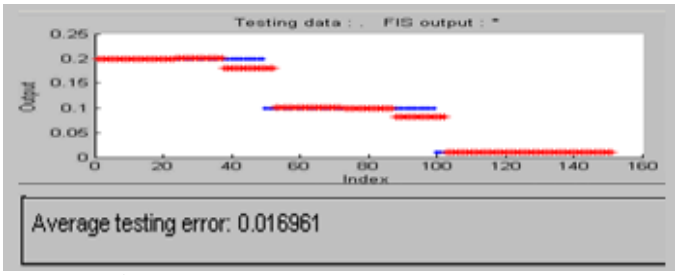

Fig. 8 Result for Framework2 with 5 input MFs

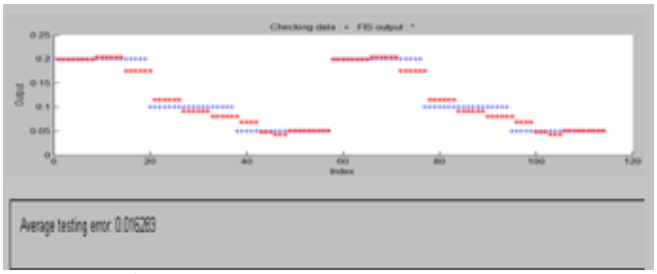

Fig. 10 Result for Framework4 with 3 input MFs

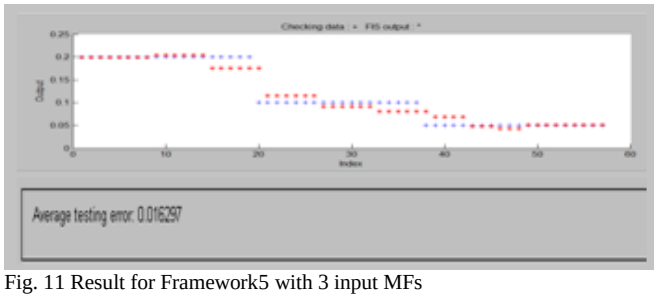

Fig. 11 Result for Framework5 with 3 input MFs

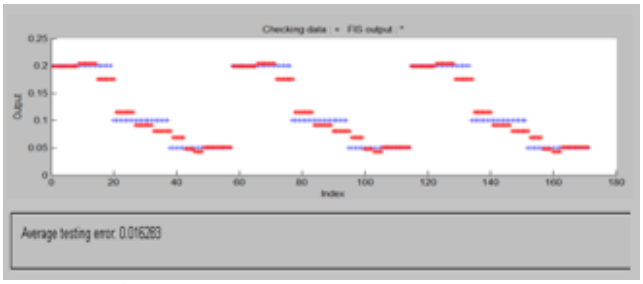

Fig. 9 Result for Framework3 with 3 input MFs

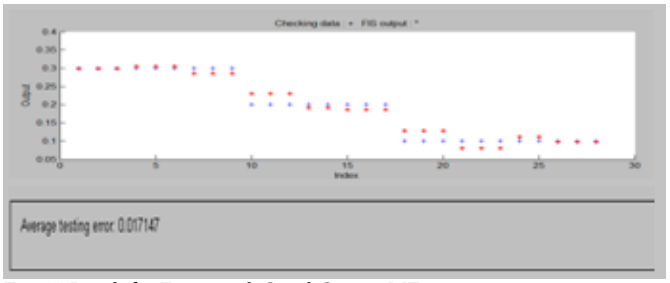

Fig. 12 Result for Framework 6 with 3 input MFs

A. Comparisions

In the previous sections, we presented the six extensive experiments conducted for parameter tuning framework for phishing website detect. The techniques and the previous results are compared to determine the best results. Individual framework was assigned different sizes of training and testing sets, different numbers of input and output membership functions, parameter optimization methods, different numbers of epochs and numbers of error tolerance. Framework3 and Framework4 obtained similar outputs. Although the frameworks 3, 4 and 5 performed better with 98.4%, they all have small differences in average error rates. Thus framework5 outperformed with a small average error rates. The difference is 0.000014 as shown in Table III.

In terms of previous work results, our work is not directly comparable for the following reasons: Firstly, our work has considered all possible sources. These sources include legitimate site rules, user-behaviour profile, PhishTank, user-specific site, pop-up windows and user's credential profile together with existing relevant journals. Secondly, from those sources 342 comprehensive features were gathered that were used for modeling procedures. Thirdly, we applied Neuro-fuzzy algorithm which has been used in our work but has not been used in other studies in this field. The previous work for example: Aburrous's two studies applied fuzzy logic and datamining techniques with 27 features to detect phishing websites and achieved 83% and 84.4% accuracy [2], [30]. Both Aburrous's studies suffered high false positives. From their source they only considered phishTank as a source with only 27 features which are a small size. Ma also used a similar approach to Aburrou, but with large lexical features extracted from URL only [5]. They achieved 95-99% accuracy.

These previous studies have not actually used all the possible features in terms of size and diversity, therefore our 98.4% accuracy is much stronger than the existing results. Moreover parameter tuning framework have not been condered in the literature in this field [9].

B. Findings

Based on the results of our experiment, we found that applying neuro-fuzzy algorithm with comprehensive feature-set and proper parameter tuning can enhance system performance with high accuracy in real-time. We also found that while features and parameters have influence on model performance, parameters have direct effect on model performance. The information about parameter tuning framework will provide guidance to the researchers who are using similar techniques. It will decrease difficulties and increase confidence during the process of parameters tuning.

C. Contributions

In terms of contribution, this paper introduced a parameter tuning framework for phishing detection websites based on a neuro-fuzzy algorithm, using 6 feature-sets, (2) we identified different numbers of membership functions, parameter optimization, different numbers of epochs, different sizes of feature-sets all on a single platform.

The advantage is that the framework will guide researchers who are using similar techniques to set parameters. It will decrease difficulties while increasing confidence in the process of tuning parameters for a given problem.

D. Limitations

In light of results from our extensive experiment, framework6 was outperformed, which achieved on average errors of 0.017147. This problem was due to some defective data that caused overfitting. As well as this, unrefined parameter tuning also confuses parameter that caused the model performance to suffer.

VIII. CONCLUSIONS AND FUTURE WORK

Based on Jiang and Ma's findings [9], [23], this paper has presented a novel parameter tuning framework, using neuro-fuzzy with six different feature-sets, different membership functions, number of parameters, and varied epochs. 6 experiments were carried out using 2-fold cross-validation to train and to validate the identified models. We found that proper parameter tuning with comprehensive feature-sets applying neuro-fuzzy system can improve system performance. In this paper, our main contribution includes: (1) We introduced parameter tuning framework based on a neuro-fuzzy algorithm, using 6 feature-sets, (2) we identified different numbers of membership functions, parameter optimization, different numbers of epochs, different sizes of feature-sets all on a single platform.

The advantage is that the framework will provide guidance to the researchers who are using similar techniques. It will decrease difficulties while increasing confidence in the process of tuning parameters for a given problem. The future

work will be to apply other different cross- validation methods and very large feature-sets.

REFERENCES

[1] Financial Fraud Action UK, Cheque & Credit clearing Company, UKCARDS Association. Deception crimes drive small increase in card fraud and online banking fraud losses. Press Release, pp. 2, 2012 [online] www.financialfraudaction.org.uk. Accessed 24.7.2013.

[2] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Modelling Intelligent Phishing Detection System for e-Banking using Fuzzy data Mining," International, 2009.

[3] N. Sanglerdsinlapachai, and A. Rungsawang, "Using Domain Top-page Similarity Feature in machine learning-based Web Phishing Detection," In Procedings of IEEE 3rd International Conference on knowledge Discovery and Data Mining, pp. 187-190, 2010.

[4] G. Xiang, B. A. Pendleton, J. Hong, "Modelling content from human-verified blacklist for accurate zero-hour phish detection," probabilistic approach for zero hour phish detection, In Proceedings of the 15th European, 2009.

[5] J. Ma, L. Saul, S. Savag, G. Voelker, "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs," Proc. of the 15th International Conference on Knowledge Discovery and Data Mining, Paris, France, pp. 1245-1254, 2009..

[6] PhishTank Site Checker (2013), GS! Networks, [online] < https://addons.mozilla.org/en-US/firefox/addon/phishtank-sitechecker/reviews/> Accessed 22.2.2014.

[7] C. Whittaker, B. Ryner, M. Nazif, "Large-Scale Automatic Classification of Phishing Pages," In the 17th Annual Network and Distributed System Security {NDSS'10} Symposium, 2010.

[8] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, C. Zhang, "An empirical analysis of phishing blacklists" in Proceedings of the 6thConference on Email and Anti-Spam, 2009.

[9] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system. IEEE,". Transactions on systems, MAN, and Cybernetics, Vol. 23, No. 3, 1993.

[10] R. Suriya, K. Saravanan, A. Thangavelu, "An integrated approach to detect phishing mail attacks a case study," SIN Proceedings of the 2nd international conference on Security of information and networks, north Cyprus, Turkey, October 2009, 6-10, pp. 193-199, vol. 3 ACM New York.

[11] L. Wenyin, N. Fang, X. Quan, B. Qiu, G. Liu, "Discovering Phishing Target based on Semantic Link Network," *Future Generation Computer Systems*, Elsevier, Volume 26, Issue 3, March 2010, pp. 381-388.

[12] G. Xiang, B. A. Pendleton, J. I. Hong, C. P. Rose, "A hierarchical adaptive," Symposium on Research in Computer Security (ESORICS'10). 268–285, 2010.

[13] X. Dong, J. A. Clerk, J .L. Jacob, "Defending the weakest link: Phishing Website Detection by analysing User Behaviours," IEEE Telecommun System, 45: pp. 215 – 226, 2010.

[14] B. Wardman, T. Stallings, G. Warner, A. Skjellum, "High-Performance Content-Based Phishing Attack Detection," eCrime Researchers Summit (eCrime), pp. 1-9, Conference: 7-9 Nov. 201 1, San Diego, CA.

[15] A. P. Barraclough, M. A. Hossain, M.A. Tahir, G. Sexton, N. Aslam "Intelligent phishing detection and protection scheme for online transactions," Expert Systems with Application 40, pp. 4697-4706, 2013.

[16] A. Le, A. Markopoulou, M. Faloutsos, "Phishdef: Url names say it all," INFOCOM, Proceedings IEEE, pp. 191-195, 2010.

[17] H. Huh, H. Kim, "Phishing Detection with popular search engine: Simple and effective", In Proceeding FPS'11 Proceedings of the 4th Canada-France MITACS conference on Foundations and Practice of Security, pp 194-207, 2012.

[18] D. Canali, M, Cova, G. Vigna, C. Krugel "Prophiler: A fast filter for the large-scale detection of malicious web pages," In Proceedings of the International World Wide Web Conference,. 2011.

[19] G. Xiang, J. Hong, C. P. Rose, L. Cranor," Cantina+: A feature-rich machine learning framework for detecting phishing web sites," ACM Transactions on Information and System Security (TISSEC), 14(2), pp. 2- 21, 2011.

[20] W. Ead, W. Abdelwahed, H. Abdul-Kader, "Adaptive Fuzzy Classification- Rule Algorithm in Detection Malicious Web Sites from Suspicious URLS," International Arab Journal of e-Technology 3 (1), pp. 1-9, 2013.

[21] G. Xiang , "Toward a phish free world: A feature-type-aware cascaded learning framework for phish detection", Thesis, Language technologies institute, School of computer science., 2013.

[22] Pinsent manson law expert, (2011). [online] http://www.pinsentmasons.com/en/expertise/sectors/core-industries--markets/universitiesandhighereducation/ Accessed 28.12.11.

[23] PhishTank, "Join the fight against phishing," 2011. [online] < http://www.phishtank.com/ > Accessed 5.6.2012 and 10.7.2013.

[24] Barclays Bank "online banking," 2012. [online] < http://www.barclays.co.uk/ > Accessed 8.12.2012.

[25] Financial Service Authority (FSA), (2013), UK [online] http://hb.betterregulation.com/external/List%20of%20banks%20-%2028%20February%202013.pdf and <www.fsa.gov.uk> Accessed 8.12.2012.

[26] G. B., Huange, Q. Y., Zhu, K. Z., Mao, C. K., Siew, P. Saratchandran, & N.Sundararajan, , (2006). Can threshold networks be trained directly? *IEEE Trans. Circuits syst. II*, vol. 53, no 3, 187-191.

[27] R. Kohavi, (1995). A study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *The International Joint Conference on Artificial Intelligence*, Computer Science Department Stanford University (IJCAI).

[28] J. T. Ma, "Learning to detect malicious URLs," Thesis, University of California, 2010.

[29] P. B. Sivarao, N.S.M. El-Tayeb, "A New Approach of Adaptive Network-Based Fuzzy Inference System Modelling in Laser Processing-A Graphical User Interface (GUI) Based," Journal of Computer Science. 5 (10), pp. 704-710, 2009.

[30] Complying with anti-phishing regulation (2012) http://help.wildapricot.com/display/DOC/Complying+with+anti-spam+regulations

[31] PhishTank, "Join the fight against phishing," 2012. [online] < http://www.phishtank.com/ > Accessed 5.7.2013 and 10.7.2013.

[32] M. Aburrous, M. A. Hossain, K. Dahal and F. Thabtah "Intelligent phishing detection system for e-banking using fuzzy data mining,"Expert Systems with Applications 37, pp. 7913-7921, 2010.

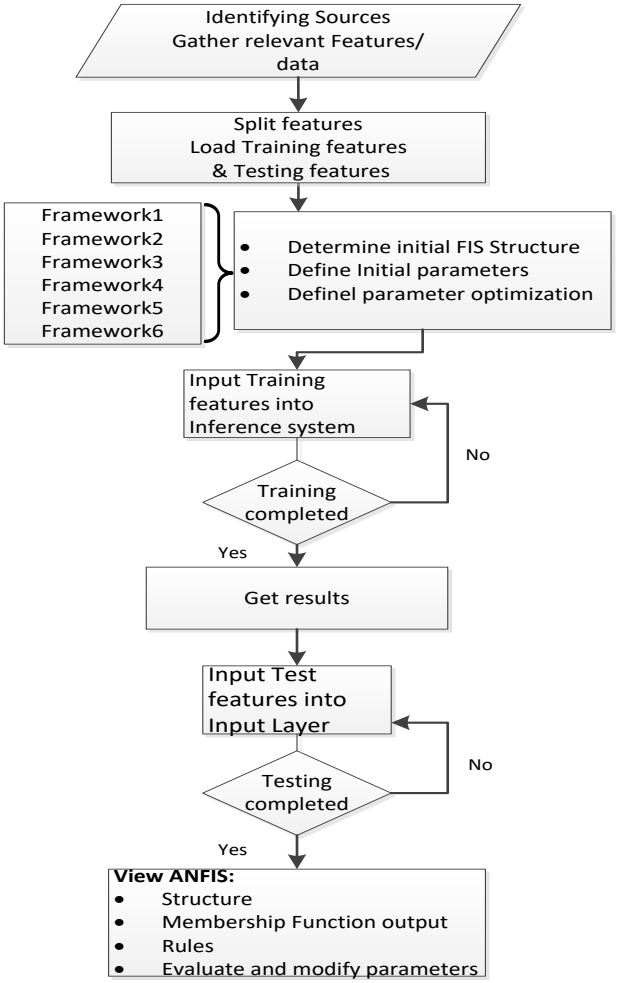

Fig. 13. Framework Process diagram