Journal of Theoretical and Applied Information Technology
15th October 2018. Vol.96. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MALICIOUS URL CLASSIFICATION SYSTEM USING MULTI-LAYER PERCEPTRON TECHNIQUE

[1]MOHAMMAD FAZLI BAHARUDDIN, [2]TENGKU ADIL TENGKU IZHAR, [3]MOHD SHAMSUL MOHD SHOID

[1,2,3]Faculty of Information Management, Universiti Teknologi MARA (UiTM),

Shah Alam, Selangor, Malaysia

E-mail: [1]fazli811@puncakalam.uitm.edu.my

ABSTRACT

Currently web-based applications, such as online shopping, education and web based discussion forums are popular. The employments of these applications have successfully assist organizations to stay competitive. Nevertheless, most of website developers are using Content Management Systems (CMS) as a platform to build a website. CMS provides third party plug-in in which this services has lack of control. CMS is designed to enable non-technical user with less knowledge on computer programming, graphics imaging tools, or markup language like HTML to develop their own website. The drawbacks that were highlighted when using CMS are software and operating systems are patched for security threats. Due to lack of patching by the user, hackers can use unpatched CMS software to exploit vulnerabilities to enter the website or web based application. This is the evident of exposing web application to cyber security risks such as malicious Uniform Resource Locator (URL). Meanwhile, malicious URL is the URL of a website that attempts to do the illegal activities on the client side. Furthermore some malicious URL can embed the malicious scripts into the web pages and exploit the vulnerabilities when the user browses such websites. This study is a baseline of measuring the effectiveness of identifying Malicious URLs by using Multi-Layer Perceptron Technique. The identification of malicious URL will be beneficial to web developer to improve the security of web based application. It is then assisting the user or organizations to access the websites without hesitate and doubt. By providing the necessary result and outputs of the effectiveness, this information can be used either for future research of any machine learning techniques.

Keywords: *Malicious URL Classification, Multi-Layer Perceptron, Content Management System, Web-based Application, Web Vulnerability*

1. INTRODUCTION

Websites application is reported widely used in online services [1]. In online services many website developer use Content Management Systems (CMS) to build the website. CMS provides third party plug-in in which this, services has lack of control. CMS is designed to enable non-technical user with less knowledge on computer programming, graphics imaging tools, or markup language like HTML to add new information to websites [2]. One of the drawbacks that were highlighted when using CMS is software and operating systems are patched for security threats. Due to lack of patching by the user, hackers can use unpatched CMS software to exploit vulnerabilities to enter the website or web –based application. This is the evident of exposing web application to cyber security risks such as malicious Uniform Resource Locator (URL). Malicious URL is the URL of a website that attempts to do the illegal activities on the client side. For example, website asks the clients to download the malicious files without any permission. This action would create system vulnerabilities or turn the client side into the virus attack. Furthermore some malicious URL can embed the malicious scripts into the web pages and exploit the vulnerabilities when the user browses such websites.

Another security issues where malicious URL can occur is the use of hundred percent hand-coded. Many website developers use hundred percent hand- coded always think of it is safe, however it is still difficult that a special character is not sanitized or web developers are not aware of a new attacking technique [3]. In new era of information communication and technology, it is really hard to guaranty that the developed website is safe without

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

tests on its security from threats such as malicious URL. In addition, today's Web applications can also contain dangerous security flaws.

Many artificial intelligence techniques have been used to detect malicious URL such as Decision Tree, Support Vector Machines, Naive Bayes [4]. These techniques have different performance. Support Vector Machines is the most appropriate classifier for classification of malicious code such as SQL Injection and XSS [4].

In this research Multi-Layer Perceptron (MLP) technique has been used to detect the malicious URL. Multi-Layer Perceptron is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate output [5]. This technique has been selected because particularly effective for predicting malicious URL when the networks have a large dataset. Otherwise in previous research it has been mentioned that MLP technique has high accuracy rate [1].Thus, this study applies MLP technique to detect malicious URL.

2. LITERATURE REVIEW

Almost all information systems and business applications such as e-commerce, banking, transportation, and blogs are now built as web-based database applications. However in recent years Web Vulnerabilities have become major threat to computer systems security. The most common website vulnerabilities attacks are SQL injections, cross-site scripting, and HTTP splitting attacks. In 2010 CWE/SANS top 25 most dangerous software errors report were published [6]. The two top positions are taken by web malicious codes are Cross Site Scripting (XSS) and SQL Injections Attack (SQLIAs). The purpose of SQLIAs is to control databases in a web application illegally. SQLIAs enable attacker to steal or change important data (e.g. addresses, and credit card number, etc.) in databases [7]. Meanwhile the purpose of XSS is to inject malicious scripts in a web application. When users access the web application which is injected XSS, injected scripts do bad things on the user's web browser (e.g. phishing, and stealing cookies, etc.)

All of the attack becomes risks in web environment and most dangerous to website developers because of the damage they may cause to the victim business. By inputting malicious code that can attack vulnerabilities, it enables one to perform various illegal acts, such as SQL Injection Attack and Cross Site Scripting [1]. All these applications are exposed to attack that any existing security vulnerability will most probably be uncovered and exploited, which may have a highly negative impact on users. In general the web malicious code attacks were disrupted or damage the web applications.

2.1 Causes of Vulnerabilities

All vulnerabilities are identified in Web applications, problems caused by unchecked input are recognized as being the most common [8]. To exploit unchecked input the attackers needs to achieve two goals which is Inject malicious data into Web applications and Manipulate applications using malicious data.

2.2 SQL Injection Problem Statement

The SQL Injection attack will affects the security of personal, social, financial and legal information for both individuals and organizations [9]. When the hacker changes the semantic or syntactic logic of a SQL text string by s string by inserting SQL keywords or special symbols within the original SQL command that will be executed at the database layer of an application, a SQL injection attack will takes place.

From the previous researches, SQL injection attacks occur when user input variables are not strongly types, thus making them vulnerable to attack [10]. As a result, these attacks can produce unauthorized handling of data, retrieval of confidential information, and in the worst possible case, taking over control of the application server. Traditional security mechanisms such as firewalls or IDSs are not very efficient in detecting and preventing these types of attacks.

Besides that one of the major problems with SQL injection is the various forms of vulnerabilities that exist. Based on previous proposals for a SQL injection are incorporating artificial intelligence and hybrids systems. In other cases, artificial intelligence techniques have been applied to face the SQL injection attack likes Web Application Vulnerability and Error Scanner (WAVES) use a blackbox technique [9]. This technique includes a machine learning approach.

2.3 SQL Injection Example

SQL injections are caused by unchecked user input being passed to a back-end database for execution. The hacker may embed SQL commands into the data he sends to the application, leading to unintended actions performed on the back-end database. When exploited, a SQL injection may

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

cause unauthorized access to sensitive data, updates or deletions from the database, and even shell command execution as shown in Figure 1.

```
A simple example of a SQL injection is shown
below:
HttpServletRequest request = ...;
String userName = request.getParameter("name");
Connection con = ...
String query = "SELECT * FROM Users " +
" WHERE name = '" + userName + "'";
        con.execute(query);
```

Figure 1: SQL Injection Example

2.4 Machine Learning Techniques

In previous works by [11], has discussed that machine learning techniques which is appropriate used for web malicious code. Machine Learning method, usually considered as the application of machine learning which used huge data sets, has already been used in efforts to detect and protect against malicious codes. In this previous paper also discussed which technique is better for identifying malicious URL. Most of them do the experiment by using blackbox techniques. Results show that classification using machine learning techniques could classify website vulnerabilities to 99% accuracy [1]. From the error detection perception with different method the Anomaly Agent and Bayesian methods provide the better results.

Most machine learning work has focused on detecting technical attacks that originate from outside a particular network or system. This is actually a very small part of the security space. The ideas above touch on some aspects of security that seem to have appropriate data available, but that do not seem to have been as closely examined. There are certainly many existing and emerging areas where machine learning approaches can bring new improvements in security.

- **Multi - Layer Perceptron**

A multilayer perceptron (MLP) is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate outputs [5]. MLP utilizes a supervised learning technique called back propagation for training the network. Neural networks are different paradigm for computing and it is an inspiration from neuroscience. Neural networks are particularly effective for predicting events when the networks have a large database. This network imitates the human brain. Artificial neurons or processing elements are highly simplified models of biological neurons. As in biological neurons, artificial neurons have a number of inputs, cell body and output that can be connected to a number of other artificial neurons. This network is densely interconnected together by learning rule which to adjust the strength of the connection between the units in response to externally supplied data.

- **Support Vector Machines (SVM)**

Support vector machines (SVMs) have performed well on traditional text classification tasks. The method produces a linear classifier, so its concept description is a vector of weights, w, and an interceptor a threshold, b. However, unlike other linear classifiers, such as Fisher's, SVMs use a kernel function to map training data into a higher dimensioned space so that the problem is linearly separable? It then uses quadratic programming to set w and b such that the hyperplane's margin is optimal, meaning that the distance is maximal from the hyperplane to the closest examples of the positive and negative classes. During performance, the method predicts the positive class if _w· x_−b >0 and predicts the negative class otherwise. Quadratic programming can be expensive for large problems, but sequential minimal optimization (SMO) is a fast, efficient algorithm for training SVMs and is the one implemented in WEKA. Based on researchers, SVM classifier is the most appropriate classifier among classifiers [12].

- **Naive Bayes**

Naive Bayes is a probabilistic method that has a long history in information retrieval and text classification. It stores as its concept description the prior probability of each class, P(Ci), and the conditional probability of each attribute value given the class, P(vj |Ci). It estimates these quantities by counting in training data the frequency of occurrence of the classes and of the attribute values for each class. Then, assuming conditional independence of the attributes, it uses Bayes' rule to compute the posterior probability of each class given an unknown instance, returning as its prediction the class with the highest value.

- **Decision Tree**

It is a common way to organize classification schemes. Every decision tree begins with the root node, and it is considered as a "parent" for other

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

node. Each node in the tree evaluates an attribute to determine which path it should follow. Typically, the decision test is performed based on comparing a value against some constant. Classification using decision tree is performed by routing from the root node until it reaches a leaf node. These classification methods are a classic way to represent the data from a machine learning algorithm, which offers a fast and powerful way to express the structures in data [11].

- **Random forest**

A random forest which introduced by [13] is a combination of decision trees in which each tree depends on the values of a random vector sampled independently and equally distributed for all trees in the forest. In this method, after generating a large number of trees, each one votes for one class of the problem. Then, the class with more number of votes is chosen.

- **K – nearest neighbors (IBK)**

The IBK is an instance-based learning algorithm [7]. Such method, derived from the k-nearest neighbors (KNN) classifier, is a non-incremental algorithm and aims to keep a perfect consistency with the initial training set. On the other hand, the IBL algorithm is incremental and one of its goals is maximizing classification accuracy on new instances (D. W. Aha et.al, 1991). As well as in the KNN, in IBK, the classification generated for the sample i is influenced by the outcome of the classification of its k-nearest neighbors, because similar samples often have similar classifications [7][10].

- **Adaptive boosting (AdaBoost)**

The adaptive boosting [14] is a boosting algorithm widely used in pattern classification problems. In general, as any boosting method, it makes a combination of classifiers. However, it has some properties that make it more practical and easier to implement than the boosting algorithms that preceded it. One of these properties is that it does not require any prior knowledge of the predictions achieved by weak classifiers. Instead, it adapts to the bad predictions and generates a weighted majority hypothesis in which the weight of the each prediction achieved by weak classifiers it is a function of its prediction.

- **Bagging**

The bagging is a method for generating multiple versions of a classifier that are combined to achieve an aggregate classifier [15]. The classification process is similar to the boosting methods, but according to Witten and Frank (Witten et.al, 2005), unlike what occurs in the second one, in the bagging, the different models of classifiers get the same weight in the generation of a prediction.

- **LogitBoost**

The LogitBoost method [15] is a statistical version of the boosting method and, according to Witten and Frank, it has some similarities with Adaboost, but it optimizes the likelihood of a class, while the Adaboost optimizes an exponential cost function. Besides that [16] has defines this method as an algorithm for assembly of additive logistic regression models.

2.5 Related Works

Web Applications becoming demanding and popular source of entertainment, communication, work & education because make life more convenient and flexible. Web services also become so widely exposed that any existing security vulnerabilities will most probably be uncovered and exploited by hackers. This section describes related works on detecting suspicious URLs within the different technique or method.

The first previous study by [4] has approach the way to detect Malicious Web Sites from suspicious URL by using automated URL classification, using statistical methods to discover the tell-tale lexical and host-based properties of malicious Web site URLs. Based on this study, these methods are able to learn highly predictive models by extracting and will automatically analyze tens of thousands features potentially indicative of suspicious URLs. This paper used the data sets from two sources which are DMOZ Open Directory Project and the second source of benign URLs was the random URL selector for Yahoo's directory. By using this approach method the resulting classifiers obtain 95-99% accuracy, detecting large numbers of malicious Web sites from their URLs. However, not all the malicious sites are in the blacklisted.

The second previous research by [12], they conducted a feasibility study on detecting malicious web sites. The main contribution of this paper is proposed a new strategy which is detecting malicious websites from suspicious URLs based on privacy preservation. In this paper they use Singular Decomposition technique to protect the private information such as WHOIS properties. The malicious sites data are getting from Phish-Tank and Spam-scatter. The dataset include WHOIS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

properties, IP address properties, Domain name properties and Geographic properties.

Meanwhile the other previous study by [14], propose the filtering mechanism based on Multi-view analysis in order to reduce the impact from URL obfuscation techniques.

The other previous research by [6] has conducted study How to detect Malicious Web Code by using Machine Learning. It has many types of machine learning. So for this paper the researcher has investigated the aptitude of each machine learning algorithm. In this paper they are explained the background of two malicious web codes which are SQL injection attack (SQLIAs) and Cross Site Scripting (XSS).The datasets that use for this paper are collected with the cooperation of experts in attacks to web applications. The result of this research solves the problems using ability of characterizing of machine learning. They have proved that classifier by Support Vector Mechanism using Gaussian Kernel is the most appropriate classifier in their evaluation.

They were also conducted the same study which is to detect Malicious Web Code using Machine Learning technique [6]. However this researcher focuses on two of machine learning algorithm which is N-Gram Analysis and Support Vector Machines (SVM). In this research the method that they used to detect the malicious code by creating the classifier for a class classification. The purpose of classifier is to classify classification data that includes normal code and malicious code that contain SQL Injection and Cross Site Scripting. The classifier was developed by using Java with JDK 1.5. Same as previous research, this paper also give the same result which is SVM is the most appropriate classifier among classifiers in detection of malicious web code.

The next previous related work also use the Application of Machine Learning Technique for Intrusion Detection to monitor and analyze event that will be happen in a computer network, and to protect the resources from threats. According [4] network and system security is one of importance in the present data communication environment. By using unauthorized intrusion, hackers and intruders can access the system and cause the crash of the network and web services. These researchers created the Network Intrusion Detection System (NIDS) that can detect scan like port scan which is using Support Vector Machine. Based on their testing [4], they conclude this method can detect 95% of attack packets correctly and warning the administrator about the packets.

The other studies by [8] analysis the use Machine Learning Methods for Spam Host Detection. They presents a comprehensive performance evaluation of several well-known machines learning technique which are multilayer perceptron neural networks, support vector machines, decision trees, random forest, adaptive boosting of trees and k – nearest neighbour used to automatically detect and filter hosts that disseminate web spam. All these methods are chosen because most of these techniques have been evaluated as the best machine learning and data mining technique currently available [14]. The result of this analysis shows that this entire evaluated machine learning techniques has achieved good performance and good capability of generalization.

The other related works by [17] presents an intelligent and effective method that used Data Mining algorithms to detect e-banking phishing websites in an Artificial Intelligent technique.

Meanwhile, [15] has conducted a feasibility study on detection of phishing websites using machine learning technique. This study evaluates the various classifying algorithm by using workbench for data mining, Waikato Environment for Knowledge Analysis (WEKA), and using MATLAB. The four machines learning algorithms used for processing the feature are Naïve Bayes, SVM, KNN and J48 Decision Tree. The research paper by [9] used two data sets in this research and the data was test by using SVM classifier.

3. RESEARCH METHODOLOGY

All the data set of the URLs Reputation was downloading from the UCI Machine Learning Repository. The data containing 2.4 million URLs (examples) and 100 million features. After the data have been collected, there are two steps that need to be performed before the data are used to train: the data need to be pre-processed and need to be divided into subsets.

Data Pre-processing block in neural network appears between input and the first layer of the network and a post-processing block that appears between the last layer of the network and the output, as shown in Figure 2. Most of the network creation functions in the toolbox, automatically assign processing functions to network inputs and outputs. These functions transform the input and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the target values provide into values that are better suited for network training.

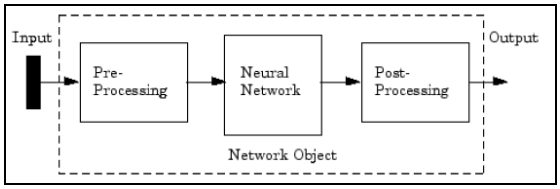

Figure 2: Data Pre-processing

System design is a process of the designing the system. This process is very important task to make sure that the implementation of the system is successfully done. While designing the system, there are several components that includes in this stage. The component of the system design is system architecture, flow chat and interface design.

a) Neural Network Toolbox

As shown in Figure 3, after the neural network classification system is designed, the graphic user interface (GUI) is roughly sketched. Matlab Neural Network Toolbox provides tools for designing, implementing, visualizing, and simulating neural networks. This is done in order to get the initial view of how the user will use the system. The GUI is designed in single forms. The GUI is used by end user to enter the data into the data panel to be process. When the end user has inserted the inputs data, result will be generated into result panel using the 'train' button. When using the Neural Network Toolbox software, batch training is significantly faster and produces smaller errors than incremental training. Result will display the type of network connection.

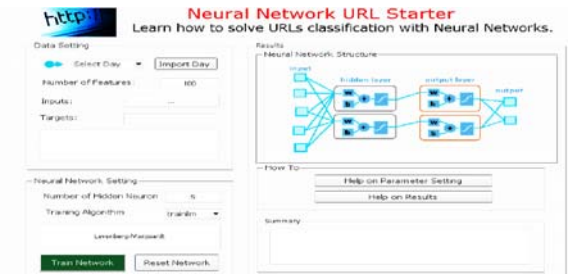

Figure 3: Neural Network Toolbox

b) Training Algorithm

For this tool it was provide 12 training algorithm can be tested. Based on feedforward artificial neural network the fastest training function is generally trainlm and it is the default training function for this feedforward neural networks. The quasi-Newton method, trainbfg is also quite fast. Both of these methods tend to be less efficient for large networks (with thousands of weights), since they require more memory and more computation time for these cases. The following table lists the algorithm that can be tested and the acronyms used to identify them.

Table 1: List of Training Algorithm

Acronym	Algorithm	Description
LM	Trainlm	Leyenberg-Marquardt
BFG	Trainbfg	BFGS Quasi-Newton
RP	Trainrp	Resilient Gradient
SCG	Trainscg	Scaled Conjugate Gradient
CGB	Traincgb	Conjugate Gradient with Powell / Beale Restarts
CGF	Traincgf	Fletcher-Powell Conjugate Gradient
CGP	Traincgp	Polak-Ribiere Conjugate Gradient
OSS	Trainoss	One Step Secant
GDX	Traingdx	Variable Learning Rate Backpropagation

4. RESULT AND DISCUSSION

The result of this experiment is measured based the number of looping/epoch that produce by the system. The best performance can show by the less number of looping/epoch while the bad performance can show by the more number of looping/epoch. For this experiment, Matlab has divides the dataset (feature-lable pair) into three smaller datasets which are training dataset, validation dataset, and testing dataset. The training dataset is used to train the neural network by adjusting the weight and bias during training stage. The validation dataset is used to estimate how good the neural network model that has been trained. The testing dataset is used to evaluate the neural network after being trained and validated. This figure describes the performance of training, validation, and testing in terms of mean squared error as the iteration (epochs) move forward. It is not easy to suggest which training algorithm will be the fastest. There are many factors that influence the training algorithm including the complexity of the problem, the number of data point in the training set, the number of weights and the error goal.

Journal of Theoretical and Applied Information Technology
15th October 2018. Vol.96. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

a) Train and Apply Multilayer Perceptron by Levenberg-Marquardt Algorithm

The first test is using Levenberg-Marquardt Algorithm (trainlm). Based on feedforward artificial neural network the fastest training function is generally trainlm and it is the default training function for this feedforward neural networks.

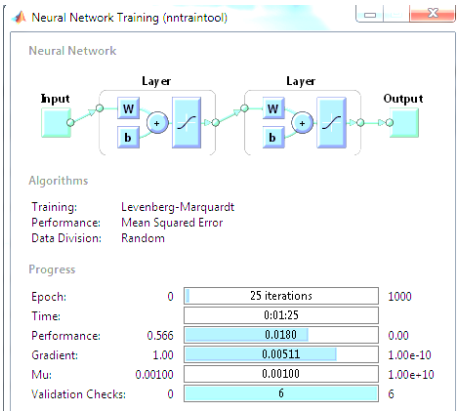

Figure 4: Test the Neural Network using Levenberg-Marquardt algorithm (trainlm)

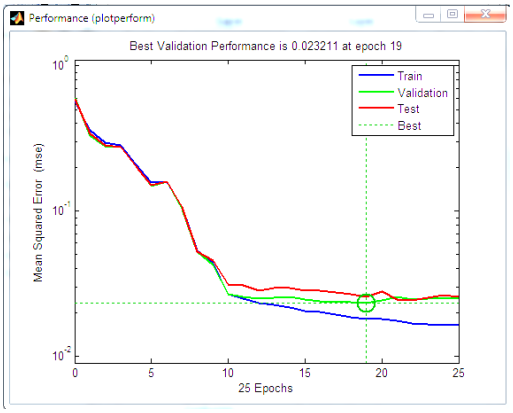

Figure 5: Mean Squared Error

From this figure we can see the best validation performance of trainlm is 0.023211 and achieved at 19 epochs. The training continued for 6 iterations before training stopped. This figure does not indicate any major problems with the training data. The graph also shown the validation and test curves are quite similar. If the result of test curve had increased significantly before the validation curve increased, then it is possible that some error might have occurred. So it means neural network can predict the minimize error if compare with the real data training.

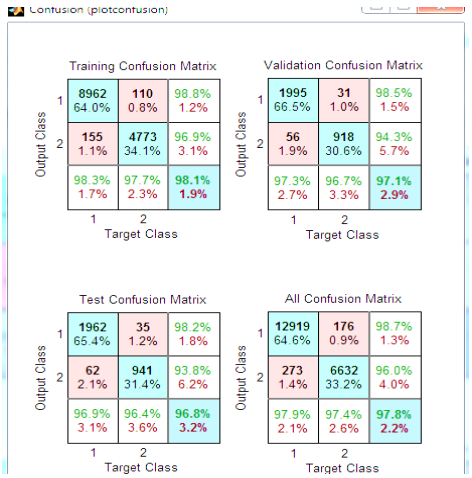

Figure 6: Neural Network Training Confusion, Epoch 25, Validation Stop

This confusion matrix describes the classification error matrix for training, testing and validation, and three kinds of data combined. It shows the various types of errors that occurred for the final trained network. The row shows the actual class (or output class that is obtained from the dataset) while the column shows the predicted class (or target class that is estimated by neural network after being trained). For this algorithm the result very accurate this achieved 97.8%.

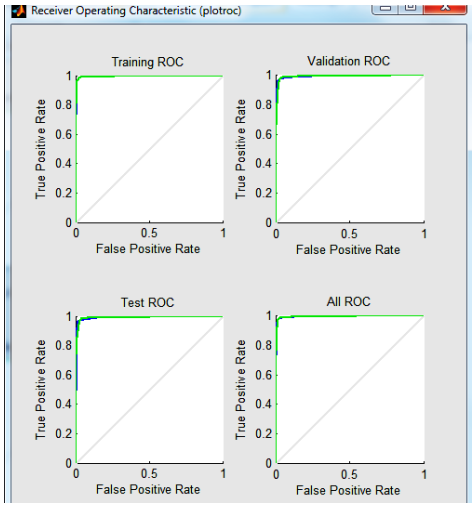

Figure 7: Neural Network Training Receiver Operating Characteristics, Epoch 25, Validation Stop

The receiver operating characteristic is a metric used to check the quality of classifiers. For each class of a classifier, roc applies threshold values

Journal of Theoretical and Applied Information Technology
15th October 2018. Vol.96. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

across the interval [0, 1] to outputs. For each threshold, two values are calculated, the True Positive Ratio (the number of outputs greater or equal to the threshold, divided by the number of one targets), and the False Positive Ratio (the number of outputs less than the threshold, divided by the number of zero targets). This graph plots the receiver operating characteristic for each output class. The more each curve hugs the left and top edges of the plot, the better the classification. So if we can see the result all the receiver operating characteristics at top edges of the plot with 100% sensitivity and 100% specificity. So it means multilayer perceptron neural network the better classification. For this algorithm, the network performs very well.

b) Train and Apply Multilayer Perceptron by BFGS Quasi-Newton

The second test is using BFGS Quasi- Newton (trainbfg). Based on feedforward artificial neural network the quasi-Newton method, trainbfg is also quite fast same as trainlm. This training stopped when the validation error increased for size iterations which occurred at iteration 23.

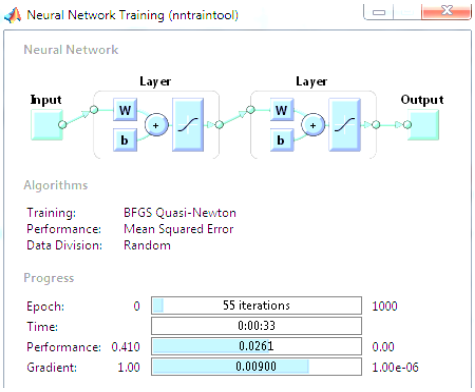

Figure 8: Test the Neural Network using BFGS Quasi-Newton algorithm (trainbfg)

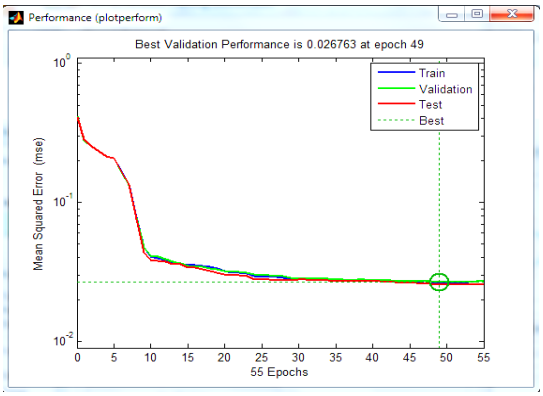

Figure 9: Mean Squared Error for trainbfg

Figure 9 shows best validation performance of BFGS Quasi-Newton algorithm is 0.026763 and achieved after 49 epochs. For this algorithm, the result is still reasonable because the final mean square error is small which 0.026763. Otherwise the test set error and the validation set error have similar curve.

5. CONCLUSION & RECOMMENDATION

This study has proven that the multilayer perceptron system is able to detect, analyze and validate the malicious URLs and produce the accurate result which is around 90-99%. It computes the given data and is able to analyze it. All of the objectives are achieved and the scope is covered which is with using data mining technique can detect and predict the malicious URL. All the methodologies are totally implemented in conducting this research. All the data are processed and the system is successfully developed by adapting neural network system model. Although all the objectives have been achieved, this system still has its disadvantages. Several improvements can be done for this project. There are:

a) The information gathered from many experts can probably increase the accuracy of this system and therefore lead to better result and reliability of the system.
b) This system can be developed further with increasing the knowledge in the data mining and improved the neural network engine in the system.
c) This system can be improved using hybrid technique to make it more accurate such as combine with bayesian technique, decision tree or support vector machines technique.

REFRENCES:

[1] Krishnaveni & Sathiykumari(2013). Multiclass Classification of XSS Web Page Attack using Machine Learning. International Journal of Computer Applications, 74(36).

[2] Nur Razia Mohd Suradi, Hema Subramaniam, Marina Hassan & Siti Fatimah Omar (2010). Development of Knowledge Portal Using Open Source Tools : A Case Study of UNISEL. International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering, 4(2).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[3] Mervat Adib Bamiah & Sarfraz Nawaz Brohi (2011). International Journal of Advanced Engineering Sciences and Technologies, 88(9).

[4] Zhang Xin Hua & Wang Zhi Jian (2010). A Static Analysis Tool for Detecting Web Application Injection Vulnerabilities for ASP Program.

[5] Sebastian Seung (2002). Multilayer Perceptron and Backpropagation learning, 1.

[6] Mattia Monga, Roberto Paleari & Emanuele Passerini, (2010). A Hybrid Analysis Framework for Detecting Web Application Vulnerabilities. (2010). ICSE 2010 Workshop, 25.

[7] A Dessiatnikoff, R. Akrout, E. Alata, M.Kaaniche & V.Nicomette (2011). A Clustering Approach for Web Vulnerabilities Detection. (2011). IEEE Computer Society, 194.

[8] Nidal Khoury, Pavol Zavarsky, Dale Lindskog & Ron Ruhl (2011). 2011 IEEE International Conference on Privacy, Security, Risk, and Trust, 1096.

[9] Cristian Pinzon, Yanira de Paz, Rosa Cano & Manual P. Rubio. (2009)An Attack Detection Mechanism Based on a Distributed Hierarchical Multi- Agent Architecture for Protecting Database, 247.

[10] Nuno Antunes & Marco Vieira (2010).Benchmarking Vulnerability Detection Tools for Web Security. 2010 IEEE International Conference on Web Services, 203.

[11] Amany Abdelhalim & Issa Traore (2000). A New Method for Learning Decision Trees from Rules. International Conference on Machine Learning and Applications, 693.

[12] Chen Junli & Jiao Licheng (2000). Classification Meachanism of Support Vector Machines. Proceedings of ICSP 2000, 1557.

[13] Andrea Avancini & Mariano Ceccato (2011). Security Testing of Web Application: Search Based Approach for Cross- Site Scripting Vulnerabilities. IEEE International Working Conference on Source Code Analysis and Manipulation, 85.

[14] Ke Wei Su, Kuo Ping Wu, Hahn Ming Lee & Te En Wei (2013). Suspicious URL Filtering based on Logistic Regression with Multi-view Analysis. 2013 Eighth Asia Joint Conference on Information Security, 78.

[15] Junho Choi, Hayoung Kim, Chang Choi, Pankoo Kim (2011). Efficient Malicious Code Detection Using N-Gram Analysis and SVM. 2011 International Conference on Network-Based Information System, 619.

[16] Yuval Elovici, Asaf Shabtai, Robert Moskovitch, Gil Tahan & Chanan Glezer (2010). Applying Machine Learning Techniques for Detection Malicious Code in Network Traffic, 46.

[17] Xiaowei Li, Yuan Xue & Bradley Malin (2012).Detecting Anomalous User Behaviours in Workflow- Driven Web Application. 2012 International Symposium on Reliable Distributed System, 1