Neuro-Fuzzy based Software Risk Estimation Tool

By Pooja Rani & Dalwinder Singh Salaria

Abstract - To develop the secure software is one of the major concerns in the software industry. To make the easier task of finding and fixing the security flaws, software developers should integrate the security at all stages of Software Development Life Cycle (SDLC). In this paper, based on Neuro-Fuzzy approach software Risk Prediction tool is created. Firstly Fuzzy Inference system is created and then Neural Network based three different training algorithms: BR (Bayesian Regulation), BP (Back propagation) and LM (Levenberg-Marquardt) are used to train the neural network. From the results it is conclude that for the Software Risk Estimation, BR (Bayesian Regulation) performs better and also achieves the greater accuracy than other algorithms.

Keywords : software security, software threat, neural network, fuzzy logic, neuro-fuzzy.

GJCST-C Classification : D.2.9
Neuro-Fuzzy based Software Risk Estimation Tool

Pooja Rani & Dalwinder Singh Salaria

Abstract - To develop the secure software is one of the major concerns in the software industry. To make the easier task of finding and fixing the security flaws, software developers should integrate the security at all stages of Software Development Life Cycle (SDLC). In this paper, based on Neuro-Fuzzy approach software Risk Prediction tool is created. Firstly Fuzzy Inference system is created and then Neural Network based three different training algorithms: BR (Bayesian Regulation), BP (Back propagation) and LM (Levenberg-Marquardt) are used to train the neural network. From the results it is conclude that for the Software Risk Estimation, BR (Bayesian Regulation) performs better and also achieves the greater accuracy than other algorithms.

Keywords: software security, software threat, neural network, fuzzy logic, neuro-fuzzy.

I. Introduction

Software systems are being used in every area to perform the different kind of activities all over the world. Due to the rapid growth of internet, technology advancement and the extensively usage of software systems results in security threats that are increasing day by day. So security becomes important concern to be considered. Threat can be any undesired event that is having potential to harm the system. Software Threat Modeling is an approach that deals with the identification, mitigation and prioritization of attacks that have to address. To predict the model for software threats, there are number of techniques like: Statistical techniques, Neural Network, Genetic Algorithm, Support Vector Machine, Fuzzy Logic and hybrid approaches: Neural Network with Genetic Algorithm, Neural Network with Support Vector Machine and Neuro-Fuzzy are being used. As it is fact that each technique has their own pros and cons. It cannot be say that one technique can overcome the limitations of all other techniques. But from the past research work, its find that the hybrid approaches provide more level of accuracy than the individual approaches.

In this paper, to create the prediction model for Software Risk, Hybrid Neuro-Fuzzy approach has been used.

Neural network based three different training algorithms: BR, BPA and LM are used.

II. Review of Literature

For Software threat prediction, various statistical approaches as well as advanced approaches are introduced in different areas where Software systems are being used. For Cyber Threat, Cyber threat trend analysis model is proposed using Hidden Markov Model (HMM), to forecast the Cyber threat trend. HMM is a tool in which hidden state is determined. After comparison with existing techniques, the proposed model provides accurate results [1]. MERIT workshop and training programs are conducted for effective training about insider threat awareness. Insider threats are those undesired events that are performed by the legitimate users [2]. Threat Analysis and Modeling (TAM) tool is used to identify the threats and evaluate the risks. This process is useful in business applications [3]. To identify the most critical large system threats, Cyber Threat Tree is implemented as directed graph known as Multiple Valued Decision Diagram (MDD). Cyber Threat Markup Language (Cyma) is used for cyber threat tree representation. Multiple Valued Decision function is used to represent the threat states and their interdependence [4].

In the area of Software Security, to identify the security vulnerabilities in software systems and to show the sequential events that occur during an attack, Regular expression based attack patterns are created. Detection of vulnerabilities is done via matching sequence of components that trigger an event during an attack [5]. Threat Mitigation, Monitoring and Management Plan (TMMMP) approach is discussed to identify the threats, to monitor the remedial measures and to deal with management plans in case of failure of remedial measures. It uses Defense In Depth (DID) strategy for threat mitigation and risk management associated with threats [6]. To identify the security flaws at early stages of software development life cycle, Extended Model Driven Architecture (MDA) approach is introduced with quantitative security assessment model. It will provide the feedback at every stage of software development life cycle [7]. To prioritize the identified threats, Common Vulnerability Scoring system (CVSS) based Risk ranking Tool is used. This tool converts Yes/No values into numerical values and then calculates the risk score using CVSS. It helps to software developer by answering the impact and exploitability of threats [8]. To overcome the limitation i.e. identification of effects by
new security threats and to developing proper countermeasures, two kind of security patterns are introduced i.e. Software Requirement Patterns (SRPs) and Software Design Patterns (SDPs). To identify the threats Software Requirement Patterns (SRPs) are used. Software Design Patterns (SDPs) are used for the identification of remedial measures against identified threats [9].

In the Networked organizations, to enhance the security by prioritizing threats and vulnerabilities, a new methodology is proposed that integrates threat modeling with formal threat analysis. This method is divided into three phases: Threat modeling, asset mapping and mitigation plan that enable the system to identify, quantify the threats and vulnerabilities [10]. For identification of threats in networked organizations, a new approach is introduced that provides reliability statistics to defense analyst to identify the top node in the network. It is useful to identify the top threats in networked organizations [11].

Now a day’s modern technique Neural Network is emerged. It is also used to model the software threats. For an intrusion detection system, user behavior modeling approach is introduced that use the neural network and support vector machine. Intrusion detection system is constructed. It is observed that the performance of this hybrid approach is superior and deliver accurate results [13]. As we know new intrusions are introduced day by day, so there is need to update the new rules to intrusion detection systems. To meet this requirement, a new intrusion detection system is presented with Genetic algorithm approach [14].

To model the real world risk scenarios, risk analysis modeling is introduced that uses fuzzy logic technique. Fuzzy logic model the vagueness in natural way. Thus it provides the accurate recommendations [15]. For electronic commerce development, web based Fuzzy Decision Support System (FDSS) is introduced. This will help to identify electronic commerce risk factors [16]. With the use of hybrid approach i.e. Neural network and support vector machine, Intrusion detection system is constructed. It is observed that the performance of this hybrid approach is superior and deliver accurate results [13]. As we know new intrusions are introduced day by day, so there is need to update the new rules to intrusion detection systems. To meet this requirement, a new intrusion detection system is presented with Genetic algorithm approach [14].

For the prediction of risk, Neuro-Fuzzy approach is used in this paper. Because the combination of Neural Network and Fuzzy Logic results in more robustness and better performance [21].

III. Neuro-Fuzzy Risk Prediction Model

For the prediction of risk, Neuro-Fuzzy approach is used in this paper. Because the combination of Neural Network and Fuzzy Logic results in such hybrid intelligent system that is having learning ability to optimize its parameters with the use of neural network and to represent the knowledge in an interpretable manner, with the use of Fuzzy System. The hybrid Neuro-Fuzzy technique is well suitable to those areas or applications, where the interpretation and interaction of user is required. Neuro-Fuzzy approach provides more accurate results than other existing hybrid techniques.

a) Fuzzy Inference System

Fuzzy Inference System is based on the concept of Fuzzy set, If Then Rules and Defuzzification. In this paper, MATLAB Fuzzy toolbox that is Graphical User Interface tool used to build the Fuzzy Inference System. To determine how Neuro-Fuzzy approach can be applied to evaluate the Software risk, some of the software factors that affect the security vulnerability are considered. These risk factors are abstracted from [22] [23] [24]. Regarding these input attributes, Corresponding security vulnerability output in the form of Low, Medium, High, Very Low and Very High are obtained from Software industry experts in from of surveys. The total 17 input risk attributes includes the following.

1. Faulty/Changing Requirements.
2. Lack of user Co-operation.
3. Poor Project Planning.
5. Undefined Project Milestones.
7. Insufficiently Trained Team Members.
8. Lack of Specialization.
9. Inexperienced Project Manager.
10. Schedule variation.
12. Deviation From Software Requirements.
15. Limitations on Real Time Performance Activities or Tasks.
17. Wrong Functions, Properties and UI(User Interface) Development.
i. Fuzzification

Fuzzification is the process to describe the input parameters through linguistic variables with meaning like ‘Low’, ‘High’, ‘Medium’, ‘Very Low’ and ‘Very High’. Fuzzy sets are representation of input parameters. These sets are represented by Membership Functions. Input parameters are represents by Zmf (Z-shaped built-in membership function). Similarly, Output parameters are represented by Gauss (Gaussian curve built-in membership function).

ii. Rule Evaluation

The total 137 if-then rules are generated after the creation of input output fuzzy sets and Membership functions. In the rules ‘T’ means “True” and representing value 1 and ‘F’ means “False” and representing value 0. The rules created in rule base of Fuzzy Inference System (FIS) are represented in the following format:

If(Fault/Changing Requirements is ‘T’) and (Lack of user Co-operation is ‘F’) and (Poor Project Planning is ‘T’) and (Poor Project management and Resource Estimation is ‘F’) and (Undefined Project Milestones is ‘T’) and (Personnel Shortfalls is ‘F’) and (Insufficiently Trained Team Members is ‘T’) and (Lack of Specialization is ‘F’) and (Inexperienced Project Manager is ‘T’) and (Schedule variation is ‘F’) and (Budget variation is ‘T’) and (Deviation From Software Requirements is ‘F’) and (Shortfalls in Externally Furnished Components is ‘T’) and (Shortfalls in Externally Performed Tasks is ‘F’) and (Limitations on Real Time Performance Activities or Tasks is ‘T’) and (Computer Science Difficulties is ‘F’) and (Wrong Functions, Properties and UI Development is ‘T’).

iii. Defuzzification

Defuzzification is the process to calculate the output, after applying if-then rules. It refers the way in which fuzzy sets are transformed into numerical value. Seventeen Input Parameters and Output parameter named Security Vulnerability are represented in Fig 1. Fuzzy Inference System Editor is used to achieve this representation.

Figure 1 : Using FIS Editor Input and Output Parameters Representation

For a given set of input parameters like [Faulty/Changing Requirements Lack of user Co-operation Poor Project Planning Poor Project management and Resource Estimation Undefined Project Milestones Personnel Shortfalls Insufficiently Trained Team Members Lack of Specialization Inexperienced Team Members Schedule variation Budget variation Deviation From Software Requirements Shortfalls in Externally Furnished Components Shortfalls in Externally Performed Tasks Limitations on Real Time Performance Activities or Tasks Computer Science Difficulties Wrong Functions, Properties and UI Development] say [1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0]. Rule Viewer is used to see the output of Security Vulnerability i.e. generated 0.5 is specified at the top of graph corresponding to considered set of input variables in Fig 2 shown below.

Figure 2 : Security Vulnerability Generation in Rule Viewer

b) Neural Network Architecture

After completing the work of Fuzzy System now next step to move on to Neural Network. In this paper Neural Network based three different algorithms are used: Levenberg-Marquardt (trainlm), Back propagation algorithm, and Bayesian Regulation.

Levenberg-Marquardt (trainlm) is a network training function that according to Levenberg-Marquardt optimization updates its weight and bias values. It is fastest algorithm. Limitation of Levenberg-Marquardt algorithm is that it consumes more memory.

Back propagation (traindx) is a learning algorithm means it learns from many inputs for desired output. It is very simple. It does not require any specialization. But the Limitation of this algorithm is that its having low prediction capabilities. Due to low prediction capabilities, it does not provide accurate results.

Bayesian Regulation (Trainbr) is advanced method. This algorithm is more suitable for those prediction cases where large number of inputs is used to predict the output. Many researchers has used
Liebenberg-Marquardt and Back-propagation algorithm for training phase.

IV. EXPERIMENTAL ANALYSIS

A feed-forward network with three different training algorithms: BR, BPA and LM are used. 12 neurons for input layer, 12 for hidden layer and 1 for output layer are used for the implementation of Neural Network.

a) Source of Training Data

As it above discussed that after generating the fuzzy rules, output is generated corresponding to fuzzy set of input variables. This training data is used to train the neural network.

b) Tool Development

For the prediction of Risk, Risk development tool is generated using MATLAB. As three different algorithms BR, BPA and LM are used so three different Graphical User Interfaces are created. Firstly Using BR algorithm GUI (Graphical User Interface) is created and shown below in fig 3.

Secondly GUI (Graphical User Interface) is created by using BPA as shown below in fig 4.

Finally 3rd GUI (Graphical User Interface) is created by using LM algorithm as shown below in fig 5.

V. RESULTS AND COMPARISON

Neural Network is trained with three different algorithms: BR, BPA & LM and outputs are obtained. From the table 1. The comparison among three different algorithms can be seen. In the table 17 inputs parameters are used and corresponding Security vulnerability output is computed for BR, BP and LM algorithms. The comparison shows that BR provides the better results than BP and LM algorithms. The results provides by BR are accurate where as BP and LM are over fitting the values for the same dataset.

The table1: Summarizes the results achieved by these three different algorithms over the same dataset. Some short terms are used in the table for input parameters are as follows.

2. LUC : Lack of user Co-operation.
3. PPP: Poor Project Planning.

![Figure 3: Neuro- Fuzzy based Software Risk Prediction Tool using BR](image)

![Figure 4: Neuro- Fuzzy Based Software Risk Prediction Tool using BP Algorithm](image)

![Figure 5: Neuro- Fuzzy based Software Risk Prediction Tool using LM Algorithm](image)
5. UPM: Undefined Project Milestones.
7. ITTM: Insufficiently Trained Team Members.
8. LOS: Lack of Specialization.
9. IPM: Inexperienced Project Manager.
10. SV: Schedule variation.
11. BV: Budget variation.
12. DFSR: Deviation From Software Requirements.
13. SEFC: Shortfalls in Externally Furnished Components.
15. LRTPA: Limitations on Real Time Performance Activities or Tasks.
16. CSD: Computer Science Difficulties.
17. WFPUID: Wrong Functions, Properties and UI (User Interface) Development.
18. Regarding Security vulnerability Output the following short terms are used.

Table 1: Risk Estimation by using Three Different Training Algorithms

<table>
<thead>
<tr>
<th>FR</th>
<th>LC</th>
<th>PP</th>
<th>MR</th>
<th>U</th>
<th>PS</th>
<th>IT</th>
<th>M</th>
<th>LOS</th>
<th>IP</th>
<th>M</th>
<th>SV</th>
<th>BV</th>
<th>DFSR</th>
<th>SE</th>
<th>FC</th>
<th>SEPT</th>
<th>LRTPA</th>
<th>CSD</th>
<th>WF</th>
<th>BR</th>
<th>SV</th>
<th>BP</th>
<th>SV</th>
<th>LM</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>94.49%</td>
<td>122.00%</td>
<td>107.39%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>92.90%</td>
<td>126.42%</td>
<td>116.41%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>89.35%</td>
<td>120.54%</td>
<td>89.61%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>84.77%</td>
<td>145.45%</td>
<td>96.73%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>70.83%</td>
<td>120.94%</td>
<td>72.35%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>72.08%</td>
<td>98.22%</td>
<td>73.25%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>68.96%</td>
<td>96.64%</td>
<td>78.46%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>65.91%</td>
<td>97.38%</td>
<td>47.59%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>59.59%</td>
<td>94.81%</td>
<td>61.16%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>53.03%</td>
<td>97.25%</td>
<td>37.60%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>47.27%</td>
<td>100.01%</td>
<td>68.02%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>42.63%</td>
<td>93.53%</td>
<td>32.06%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>37.41%</td>
<td>94.94%</td>
<td>29.79%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>33.26%</td>
<td>14.34%</td>
<td>18.33%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>28.80%</td>
<td>19.98%</td>
<td>27.09%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>21.32%</td>
<td>20.63%</td>
<td>31.52%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>17.35%</td>
<td>26.50%</td>
<td>23.16%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>12.64%</td>
<td>35.06%</td>
<td>12.81%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>5.96%</td>
<td>21.32%</td>
<td>2.85%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>2.25%</td>
<td>10.80%</td>
<td>4.48%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

VI. Conclusion

Software Risk Prediction is one of the most important tasks for the development of secure and reliable system. It should be preferred that during the early stages of software development life cycle to find and fix the security flaws. Neuro-fuzzy approach based risk prediction tool is developed using MATLAB. After creation of Fuzzy Inference System, Neural Network is trained with three different algorithms using ’triangbr’, ’traindx’ and ’trainlm’. From the results it is concluded that BR algorithm performs better than other algorithms. With the use of BR algorithm better accuracy level is achieved then other algorithms.

References