Global Journal of Computer Science and Technology
Software & Data Engineering
Volume 13 Issue 6 Version 1.0 Year 2013
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Neuro-Fuzzy based Software Risk Estimation Tool

By Pooja Rani & Dalwinder Singh Salaria
Lovely Professional University, Punjab

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*

Strictly as per the compliance and regulations of:

Neuro-Fuzzy based Software Risk Estimation Tool

Pooja Rani [α] & Dalwinder Singh Salaria [σ]

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.

Generalterms : *software risk prediction.*

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.

Author α : Student, Department of CSE Lovely Professional University Phagwara, Punjab – 144411. E-mail : erpoojapuri88@gmail.com

Author σ : Assistant Professor, CSE Dept Lovely Professional University Phagwara, Punjab – 144411. E-mail : ds_salaria@yahoo.com

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 Logic function is used to represent the threat states and their interdepend -ence [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. Identification 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 algorithm and provides better results than existing results [12]. 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].

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 ecommerce risk factors [16]. With the use of fuzzy logic secure software system (SSS) approach is introduced. It will help to avert the failed state of the system [17].

For the development of marketing strategy, hybrid intelligent system is developed with the combined approach of Neural Network, Fuzzy Logic and expert system. For the settlement of marketing strategy, this hybrid system is useful to produce intelligent advice [18]. Neural fuzzy scheme is proposed for the development of Direction of arrival (DOA) estimation algorithm by Self-constructing Neural fuzzy Inference Network (SONFIN). The performance of this newly developed algorithm is superior than RBFN [19]. To calibrate the conversion ratios for backfiring technique, calibrated model is generated by using neuro-fuzzy approach. From this model, it is concluded that higher accuracy is achieved for software size estimation [20]. To make the decision about Distributed Intrusion Prediction and Prevention system (DIPPS) , a model named Hierarchical Neuro-Fuzzy Online Risk Assessment(HiNFRA) using Neuro-fuzzy approach is introduced. This model by using Neuro-fuzzy approach 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.
4. Poor Project Management and Resource Estimation.
5. Undefined Project Milestones.
6. Personnel Shortfalls.
7. Insufficiently Trained Team Members.
8. Lack of Specialization.
9. Inexperienced Project Manager.
10. Schedule variation.
11. Budget variation.
12. Deviation From Software Requirements.
13. Shortfalls in Externally Furnished Components.
14. Shortfalls in Externally Performed Tasks.
15. Limitations on Real Time Performance Activities or Tasks.
16. Computer Science Difficulties.
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.

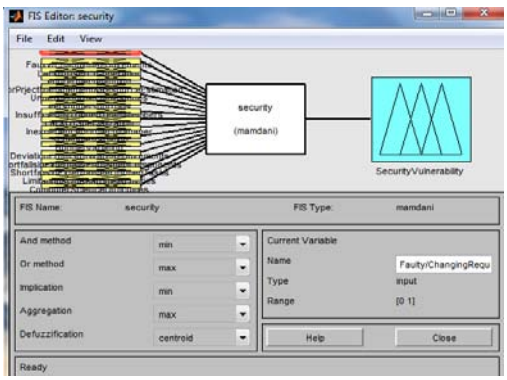

Figure 1 : Using FIS Editor Input and Output Parameters Representation

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 Project Manager 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}. 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.

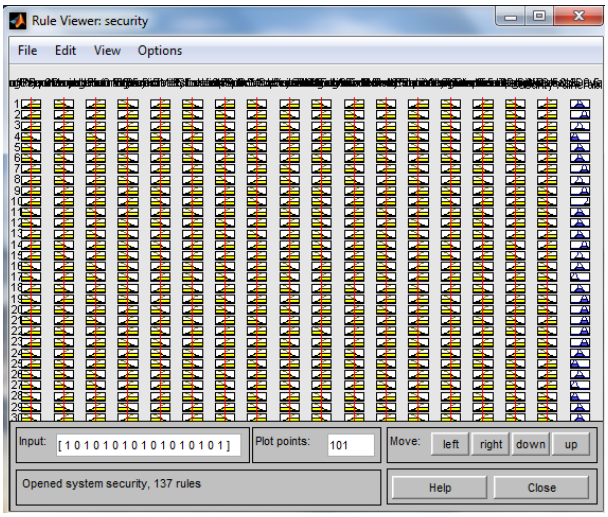

Figure 2 : Security Vulnerability Generation in Rule Viewer

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 (triangdx) 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.

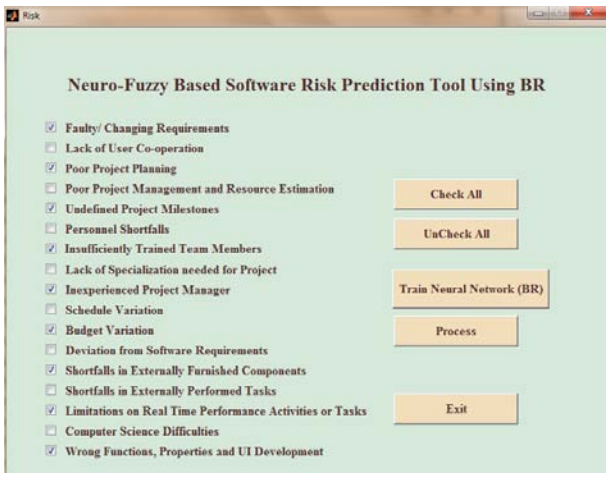

Figure 3 : Neuro- Fuzzy based Software Risk Prediction Tool using BR

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

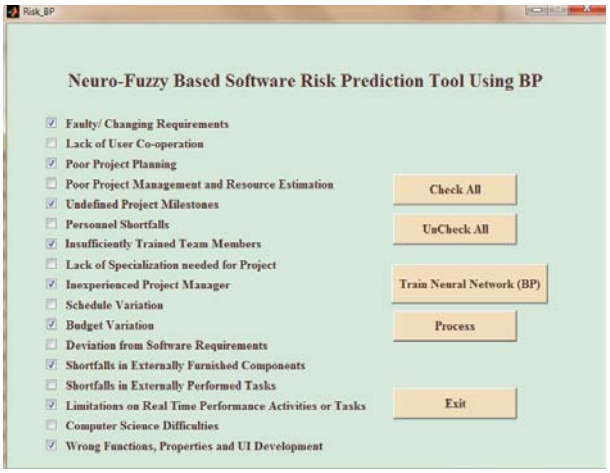

Figure 4 : Neuro- Fuzzy Based Software Risk Prediction Tool using BP Algorithm

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

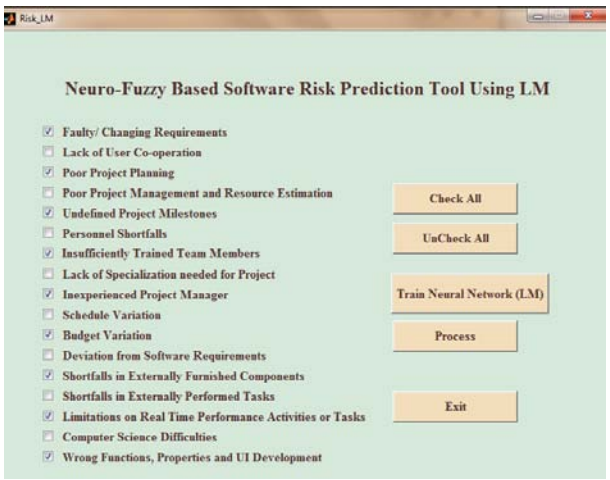

Figure 5 : Neuro- Fuzzy based Software Risk Prediction Tool using LM Algorithm

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.

1. FR : Faulty/Changing Requirements.
2. LUC : Lack of user Co-operation.
3. PPP: Poor Project Planning.

4. PPMRE: Poor Project Management and Resource Estimation.
5. UPM: Undefined Project Milestones.
6. PS: Personnel Shortfalls.
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.
14. SEPT: Shortfalls in Externally Performed Tasks.
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.
19. SVBR: Security Vulnerability Using BR (Bayesian Regulation).
20. SVBP: Security Vulnerability using BP (Back propagation).
21. SVLM: Security Vulnerability using LM (Liebenberg-Marquardt).

Table1 : Risk Estimation by using Three Different Training Algorithms

FR	LUC	PPP	PPMRE	UPM	PS	ITTM	LOS	IPM	SV	BV	DFSR	SEFC	SEPT	LRTPA	CSD	WFUID	SV BR	SV BP	SV LM
T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	94.49%	122.00%	107.50%
F	T	T	T	T	F	T	T	T	T	T	T	T	T	T	T	T	92.90%	126.42%	116.44%
T	F	T	T	F	T	T	T	T	T	T	T	T	T	T	T	F	89.29%	62.04%	79.91%
T	T	T	T	T	T	F	T	F	T	T	F	T	T	F	T	T	84.77%	145.45%	96.73%
T	T	F	T	T	T	T	T	T	F	T	T	F	T	F	F	T	79.81%	-20.94%	73.39%
F	T	T	F	T	F	T	F	T	T	F	T	T	F	T	T	T	72.06%	90.22%	73.43%
T	F	T	F	F	T	F	F	T	T	F	T	T	T	T	T	F	68.96%	36.64%	38.46%
F	T	T	T	F	T	F	T	F	T	F	T	T	T	F	T	F	65.51%	75.38%	47.59%
T	F	F	T	F	T	F	T	T	F	T	F	T	F	T	F	T	59.30%	34.81%	61.16%
T	T	F	F	T	F	F	T	T	F	T	F	F	T	F	T	T	53.03%	47.23%	57.09%
F	F	F	T	F	T	F	F	F	T	T	F	F	T	T	T	T	47.27%	100.01%	68.02%
T	F	T	F	T	F	T	F	F	T	F	T	T	F	T	F	F	43.63%	3.58%	23.40%
T	T	T	F	F	F	T	F	F	T	F	T	T	F	F	T	F	37.41%	54.94%	29.29%
F	T	F	F	F	F	F	T	T	F	F	F	T	T	T	F	T	33.26%	-14.34%	18.33%
F	F	F	T	F	T	T	F	T	F	T	T	F	T	F	F	F	28.80%	19.98%	27.09%
F	F	F	F	F	T	T	F	T	F	T	T	F	T	F	F	F	21.32%	20.63%	31.53%
T	F	F	F	T	T	T	F	F	F	T	F	F	F	F	F	F	17.35%	26.50%	23.10%
F	F	F	F	F	F	F	F	F	T	F	T	T	F	T	F	F	12.64%	35.06%	12.13%
F	F	F	F	F	F	F	F	F	F	F	F	T	F	T	F	F	5.96%	21.22%	2.85%
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	2.23%	10.60%	4.48%

VI. Conlcusion

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 'trianbr', 'traingdx' 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 Références Referencias

1. Lee, T., Jung, S. O., In, H. P., & Lee, H. J. (2007, August). Cyber Threat Trend Analysis Model Using HMM. In Information Assurance and Security, 2007. IAS 2007. Third International Symposium on (pp. 177-182). IEEE.
2. Greitzer, Frank L., Andrew P. Moore, Dawn M. Cappelli, Dee H. Andrews, Lynn A. Carroll and Thomas D. Hull. "Combating the insider cyber threat." Security & Privacy, IEEE 6, no. 1 (2008): 61-64.
3. Ingalsbe, Jeffrey A., Louis Kunimatsu, Tim Baeten and Nancy R. Mead. "Threat modeling: diving into the deep end." Software, IEEE 25, no. 1 (2008): 28-34.
4. Ongsakorn, P., Turney, K., Thornton, M., Nair, S., Szygenda, S. & Manikas, T. (2010, April). Cyber threat trees for large system threat cataloging and analysis. In Systems Conference, 2010 4th Annual IEEE (pp. 610-615). IEEE.
5. Gegick, Michael and Laurie Williams. "Matching attack patterns to security vulnerabilities in software-intensive system designs." In ACM SIGSOFT Software Engineering Notes, vol. 30, no. 4, pp. 1-7. ACM, 2005.
6. Gandotra, V., Singhal, A., & Bedi, P. (2009, October). Threat mitigation, monitoring and management plan-A new approach in risk management. In Advances in Recent Technologies in Communication and Computing, 2009.

ARTCom'09. International Conference on (pp. 719-723). IEEE.
7. Tang, X. & Shen, B. (2009, July). Extending Model Driven Architecture with Software Security Assessment. In Secure Software Integration and Reliability Improvement, 2009. SSIRI 2009. Third IEEE International Conference on (pp. 436-441). IEEE.
8. Dhillon, Danny. "Developer-Driven Threat Modeling: Lessons Learned in the Trenches." Security & Privacy, IEEE 9, no. 4 (2011): 41-47
9. Okubo, T., Kaiya, H. & Yoshioka, N. (2011, August). Effective security impact analysis with patterns for software enhancement. In Availability, Reliability and Security (ARES), 2011 Sixth International Conference on (pp. 527-534). IEEE.
10. Stango, A., Prasad, N. R. & Kyriazanos, D. M. (2009, June). A threat analysis methodology for security evaluation and enhancement planning. In Emerging Security Information, Systems and Technologies, 2009. SECURWARE'09. Third International Conference on (pp. 262-267). IEEE.
11. Frantz, T. L., & Carley, K. M. (2009, July). Information assurances and threat identification in networked organizations. In Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on (pp. 1-5). IEEE.
12. Debar, H., Becker, M., & Siboni, D. (1992, May). A neural network component for an intrusion detection system. In Research in Security and Privacy, 1992. Proceedings. 1992 IEEE Computer Society Symposium on (pp. 240-250). IEEE.
13. Mukkamala, S., Janoski, G., & Sung, A. (2002). Intrusion detection using neural networks and support vector machines. In Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on (Vol. 2, pp. 1702-1707). IEEE.
14. Gong, R. H., Zulkernine, M. & Abolmaesumi, P. (2005, May). A software implementation of a genetic algorithm based approach to network intrusion detection. In Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Conference on (pp. 246-253). IEEE.
15. Haslum, K., Abraham, A. & Knapskog, S. (2008, May). Hinfra: Hierarchical neuro-fuzzy learning for online risk assessment. In Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on (pp. 631-636). IEEE.
16. Ngai, E. W. T., & Wat, F. K. T. (2005). Fuzzy decision support system for risk analysis in e-commerce development. Decision support systems, 40(2), 235-255.
17. Gandotra, V., Singhal, A. & Bedi, P. (2010, April). A step towards secure software system using fuzzy logic. In Computer Engineering and Technology (ICCET), 2010 2nd International Conference on (Vol. 1, pp. V1-427). IEEE.
18. Li, S. (2000). The development of a hybrid intelligent system for developing marketing strategy. Decision Support Systems, 27(4), 395-409.
19. Shieh, C. S., & Lin, C. T. (2000). Direction of arrival estimation based on phase differences using neural fuzzy network. Antennas and Propagation, IEEE Transactions on, 48(7), 1115-1124.
20. Wong, J., Ho, D., & Capretz, L. F. (2008). Calibrating function point backfiring conversion ratios using neuro-fuzzy technique. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 16(06), 847-862.
21. Haslum, K., Abraham, A. & Knapskog, S. (2008, May). Hinfra: Hierarchical neuro-fuzzy learning for online risk assessment. In Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on (pp. 631-636). ieee.
22. Boehm, B. W. (1991). Software risk management: principles and practices. Software, IEEE, 8(1), 32-41.
23. Hu, Y., Zhang, X., Sun, X., Zhang, J., Du, J. & Zhao, J. (2010, November). A unified intelligent model for software project risk analysis and planning. In Information Management, Innovation Management and Industrial Engineering (ICIII), 2010 International Conference on (Vol. 4, pp. 110-113). IEEE.
24. Bragina, T. & Tabunshchyk, G. (2011, February). Fuzzy model for the software projects design risk analysis. In CAD Systems in Microelectronics (CADSM), 2011 11th International Conference. The Experience of Designing and Application of (pp. 335-341). IEEE.