European Journal of Molecular & Clinical Medicine
ISSN 2515-8260 Volume 7, Issue 4, 2020

Development of Fuzzy based intelligent System for Assessment of Risk Estimation in Software project for Hospitals Network

Gajanand Sharma[1], Ashutosh Kumar[2], Himanshu Sharma[3], Ashok Kumar Saini[4] *Neha Janu[5], Blessy Thankchan[6]

[1,2,3,4] Department of Computer Science & Engineering, JECRC University, Jaipur, India

[5]Computer Science and Engineering Department, Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur, India

[6]School of Computer & Systems Sciences, Jaipur National University

[1]gajanan.sharma@gmail.com, [2]ashucse007@gmail.com, [3]himanshu.manu.sharma@gmail.com, [4] shksaini@gmail.com, [5]neha123jaipur@gmail.com, [6] blessyt218@gmail.com

Abstract: One of the major concerns of software industries is to develop invulnerable and secure software. A project is successful if it is delivered within provided time constraints with the required quality. The aim of this research is to enhance the project planning phase by risk assessment at the start of software development life cycle. The proposed model is developed to accurate estimation of risk in software project. The software risk estimation model is being made using neuro-fuzzy approach which is beneficial to the project manager in the first phase of software development life cycle. This methodology is different from the other risk assessment process as it uses the vague factors of different type of risk parameters in software projects. Two main activities of project planning phase are risk assessment and effort estimation. Effort estimation is mostly influencing with the probability of risk estimation at the early phase of software development life cycle. Implement result shows that it reduces risk caused due to vague parameters in software projects. The intelligent system so developed is also capable of identifying the risk of similar kind in any software project for hospital network system.

Keywords: Software Project, Risk Contingency, Project Planning, Hospital Network, Neural Network, Risk Probability

1. Introduction

Risk is a situation which involves an exposure to danger, harm, loss and vulnerability. Risk is the future happenings which after the development become flaws. In order to mitigate risk developers should integrate security at all stage of software development life cycle. The practical planning of software project is with the help of risk management [1]. Many projects fail due to either simple problem is not addressed or small reasons for the software failure are neglected. Risk management in general means reducing uncertainty at every stage of software development life cycle such as uncertainty in customer requirement, indescribable nature of product, complexity of the process and obscure result of the project. Two main activities of project planning phase which have to be executed together are effort estimation and risk estimation. However, in software development projects effort estimation as well as risk management are often disconnected from each other. There are many effort estimation models such as SLIM model, Park and PRICE Systems PRICE-S model, Checkpoint model, Putnam and Quantitative Software Measurement, Rubin and Jensen and the SEER SEM model, the Estimacs model and Boehm and the COCOMO model. Risk arises from our idea of perspective like project failure, safety, security, legal liabilities and so on. Risks are uncertain event which

if becomes reality unwanted loss may occur. Risk arises from our idea of perspective like project failure, security, legal liabilities, and safety and so on. The primary objective of software team is to manage risk and avoid it at any cost [2]. But eventually not all risk can be avoided so contingency plan is made to control issue in effective manner. To analyze risk software team member must quantify the level of uncertainty and degree of loss. On the basis of level of uncertainty and degree of loss different categories of risks are classified

2. Literature Review

In order to choose specific objectives of research it is needed to follow a process to arrive at the conclusion of uniqueness, novelty and significance of the problem arise in a specific area /sub area. Therefore

There has been many procedures and processes proposed by the researchers to undergo through and arrive at certain conclusion of research objectives. There are various research and sub-division of each of the project management, but in this paper project risk management is taken into consideration. Negative risks can be minimized with the help of 5 ways. They are risk identification; risk planning, risk analysis, risk management, risk monitoring and control [3]. Subjective evaluation is done in the qualitative risk analysis and mathematical evaluation is done in quantitative risk analysis. During risk management, quantitative analysis is given less consideration in overall project. Risk management is very necessary during each process of project planning phase as most of the project problems are eliminated at this stage [4]. PMI's risk management provides the project manager accurate as well as entire direction to manage risk in the project environment. The training is very useful in software project design [5]. It has limitations which are based on the fact that software projects are considered as uncertain, risky and complex project as compared to other kind of projects.

3. Software effort estimation

The expert COCOMO model is considered to be the most famous effort estimation [6]. This model uses the previous knowledge of the similar project for effort estimation in software industries. However, the expert COCOMO model has a limitation that it cannot imprecise data and uncertain information. Uncertain information can be expressed in the form of linguistic variables such as very high, high, nominal, low or very low. Using fuzzy approach there are many fuzzy expert model which are taken into consideration for exact estimation of risks. Software project planning deals with the future estimation of software project [7]. In order to reduce risk in the project scope, time and budget are the primary elements which the project manager must be taken accurately before risk management [8].

3.1 Cocomo Model

Source lines of code are taken into consideration in COCOMO model, which is used for effort, cost and schedule estimation in software project development. COCOMO'81 was the first version of COCOMO model. There are three levels of this model which actually shows the effect of estimation. The first level provides the rough estimate for effort estimation, the equation made in the first level are modified in second level with the help of process multiplier, different phases of project are estimated at this level. COCOMO model is not suitable for risk estimation in software project development. Bohem's risk management: This is the oldest risk management which are used since 1980's for risk estimation and is taken into consideration in software engineering since 1981 [9]. He introduced that the risk can be minimized in software project planning phase in software development life cycle phase [10]. In 1988 spiral model was introduced by Bohem to overcome the limitations of waterfall model. However, it was very difficult for the project planner to convince the customer to use this model because it deals with high risk project and the success of the project is highly based on expertise [11]. In 1991 overall risk management proposed by Bohem was divided into risk assessment as well as risk control. Risk assessment is the process of risk judgment. From the table of software risk checklist, risk priorities are mentioned by the project manager on the basis of historical data [2].

3.2 Problem Statement

Risk estimation is an integral part of project planning phase and it also plays vital role in the success of the project. In software industries no risk model is developed on the basic of vague values till date. There are many parameters on which risk can be analyzed in a software product. Some of them are taken as primary consideration whereas some are avoided as they are vague parameters. Those vague parameters affect the production of the software industries in some or other ways. In common practice risk assessment is highly dependent on the size as well as the nature of software project. Moreover, risk estimation in software projects has some weaknesses [12]. The main weakness is that risk assessment completely depends on human expertise as well as personal judgment of the developer. This paper describes the neuro-fuzzy based risk estimation to detect the overall risk in the software project.

3.3 Fuzzy Logic

It is a computer-based approach which is used for developing system models in a linear control system when the precise definition and boundary do not exists towards each other. Tool that can manipulate those kinds of data which are imprecise and the mathematical tool is exactly fuzzy logic. It is a mechanism to embed human behavior and thinking into any of the control system[13]. Based on the previous knowledge human being can take decisions, in similar fashion fuzzy logic controller have the ability to deduce certain decisions based on the prior information provided to the system. So we can utilize such principle of human way of decision making into designing control systems.

3.4 Use of Neuro-fuzzy Approach

There are many type hybrid neuro-fuzzy approaches which used for prediction. Some of them are neuro-fuzzy using mamdani and sugenofis and another type is adaptive neuro-fuzzy inference system. But this paper is using neuro-fuzzy approach using mamdani fuzzy inference system as there is multiple input for single output. Fuzzy logic is used for determining the degree of truth value in terms of linguistic variables. Exact decision making can be done through fuzzy logic degree of risk can be measured with the help of inbuilt membership functions. As well as neural network [14] is used for the risk categories in case further risk arises in the system.

4. Personal Risk

It is the risk which deals the personal parameters of employee due to which issue may arise in software project. Basically, experience of employee plays important role in successful completion of project. Following are the previously used major factors and proposed fuzzy factors for personal risks.

Table 1: Process Risk of software project

Major factors	Fuzzy factors
Analyst capability, application experience, language and toolset experience, programmer's capability, virtual machine experience and personal continuity.	Programmers experience, Language efficiency based on programmer's qualification

Platform risk

This type of risk is associated with how volatile is the development environment. Relevant architecture is needed on which software is developed otherwise it could create many issues later on in future and rework would have to be done.

Table 2: Platform Risk of software project

Major factors	Fuzzy factors
Execution platform, architecture constraints, tool constraints, main storage constraints, software size constraints	Execution time, Database size

4.1 Reuse risk

It is related to the impact of reusing the application being developed in new projects. Reuse strategies totally depends on the kilo lines of code.

Table 3: Platform Risk of software project

Major factors	Fuzzy factors
Experiences, appropriate tools and other elements to ensure the success of the software project	Reusability risk, Software quality

Different risk estimator system can judge the risk in any software project with the help of different vague parameters which may be of any level. Its level is checked in linguistic terms such as low, moderate or high. Secondly there was no intelligent system till date which utilizes the main risk dataset to obtain the overall risk in the project. The manual processing takes lots of time and money which are inconvenient for the developers. Neural network consists of various transfer function or activation function with the help of which it connects input parameters to output parameters via different layers. Thus, soft computing is considered to be the best solution to the problems which are complex to answer with hard computing. In other words, it deals with the problems of uncertainty, imprecision and partially true to get robustness at low cost.

4.2 Overall Project Risk

Neural network as a part of soft computing is able to develop intelligent system for assessment of risk as a linguistic variable which could not be computed in hard computing. Overall risk in software project is directly proportional to the overall cost of the project. The nonlinear risk level and cost multiplier is totally dependent on the size of the project. The above equation helps the neural network in training to provide the overall target risk. Here risk categories are taken in the form of linguistic variables that is low, moderate and high as well as project size is measured in kilo lines of code (KLOC).

5. Implementation

The software projects are very critical at its planning phase and the success of the whole project is dependent on this phase. Risk is of various kinds and each risk arises in the software due to different parameters. There are also some vague parameters which also affect the project in some or other way. Those vague parameters are measured exactly in precise manner with the help of soft computing. Much research is done in this field but no one concentrated on vague parameters which could create risk in software project. There is much software which is used to implement fuzzy logic and neural network. Some of them are simulink, DSP system toolbox, parallel computing toolbox, and signal processing toolbox, statistic toolbox and many more. In this paper MATLAB is taken as toolbox for the simulation of fuzzy logic and neural network.

5.1 Neural Network

Secondly neural network is trained five different risk parameters. Training includes NASA dataset to judge whether the overall risk lies in the category of low, moderate or high.

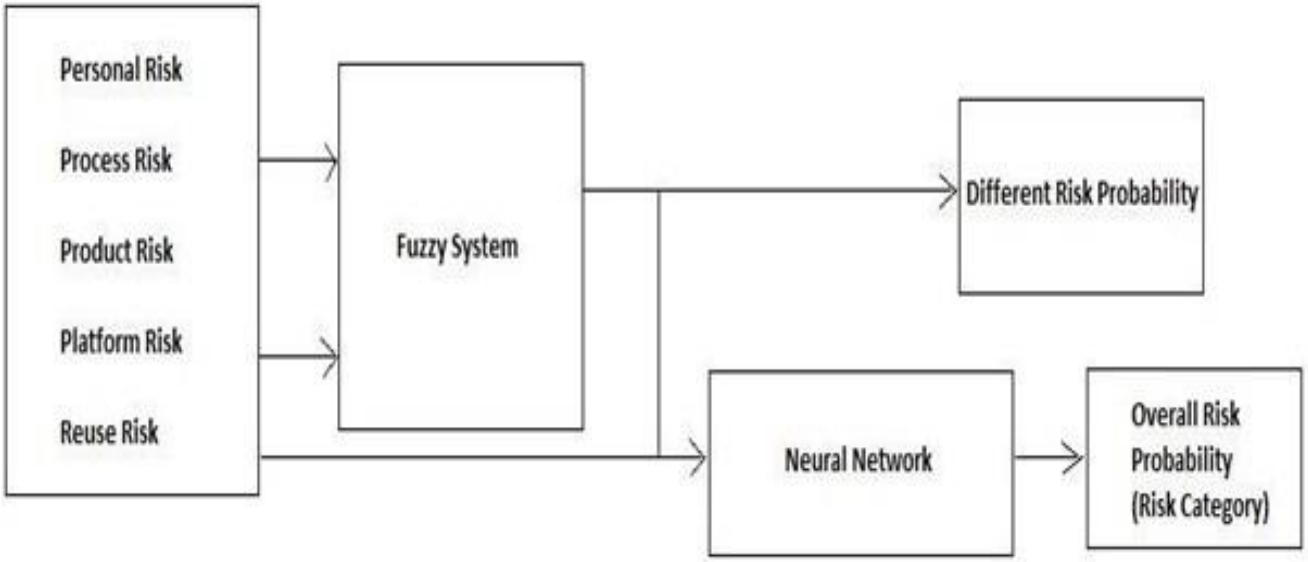

Figure 1: Flow Diagram of probability of overall risk estimation in software projects

Software project consists of various types of risk and out of which some are detectable and some are not. Much previous research has been done to measure the probability of risk contingency till date but risk caused due to vague parameters has not been taken into consideration till date. It is the responsibility of project manager to consider each and every vague parameter also for the successful completion of the software project. The risk contingency for those vague parameters are measured exactly with the help of fuzzy system. The neural network is trained with the help of NASA data set to predict the probability of overall risk contingency. The whole section of risk assessment is divided into parts which are measured with vague parameters.

4.2 Fuzzy logic Implementation

It is logic which deals with accuracy and the result so obtained is precise in Boolean logic between 0 and 1 or true and false. The membership function decides the level of accuracy and exactness in the result. The input to fuzzy system is given in the form of linguistic variables that is low, moderate, high and very high. With the help of those linguistic variables the membership function can estimate the level of risk contingency. This paper proposed the system for risk estimation using the following risk generating factors.

4.3 Neural Network Implementation

Neural network is an intelligent system having the capability to train the neurons and make it learnt with the help of different risk datasets.

Training model of neural network to classify the probability of different risk category

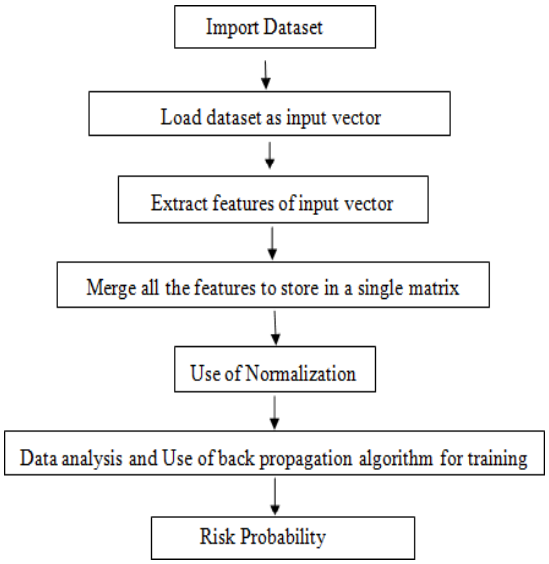

Figure 2: Risk evaluation model using Neural Network

Algorithm 1 Algorithm for training of neural network

Step 1:Load the main function and type nftool on the command window of MATLAB toolbox
Step 2: Import and choose input im train and target tmx train matrix to train the neural network
Step 3: Press next as 70 dataset is used for training, 15 dataset id used for validation and 15 dataset is used for testing.
Step 4: Provide the number of hidden neurons 10 and proceed next
Step 5: Train the network by pressing train the network.

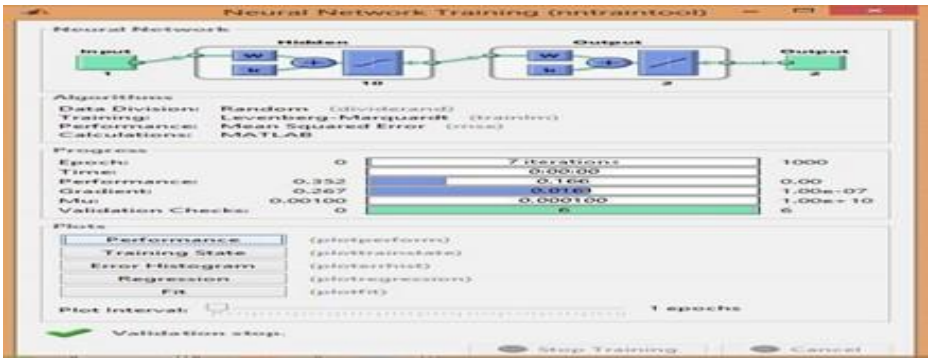

Figure 3: Train the neural network and provide learning

6. Results

The suggested software risk management framework is able to identify various forms of risk using dynamic parameters in software projects. The different risk parameters are in principle used for calculating the ultimate risk using the neural network roughly. The training offers 94.92 per cent accuracy, while the data collection observer error is 5.08 per cent. The testing takes place. The target or ultimate risk rate, which occurs in language variables in the program project, is calculated.

Figure 4: Training in neural network showing the errors in training, validation and testing

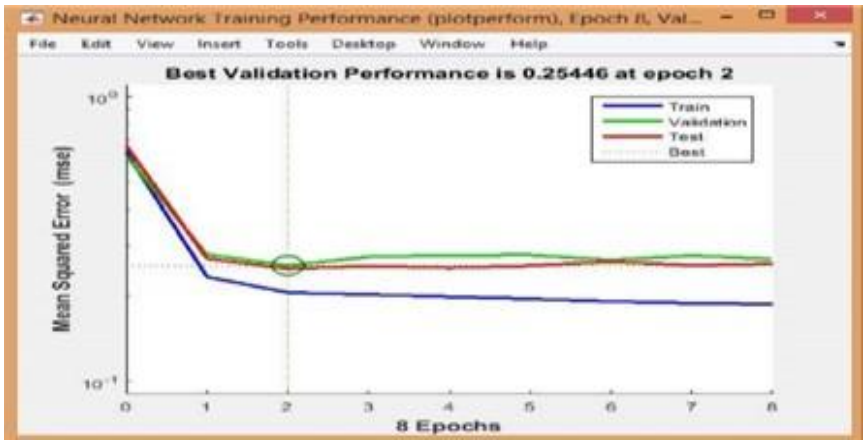

Figure 5: Performance plot after training of neural network

Performance plot show well trained risk dataset in ANN to forecast further risk in software projects. With the increase in the number of epochs, the MSE (Mean Square Error) decreases. The MSE becomes very small at the end of training which is equal to 5.0440e-1. The desired output and the obtained ANN output after training risk dataset is found to be almost same.

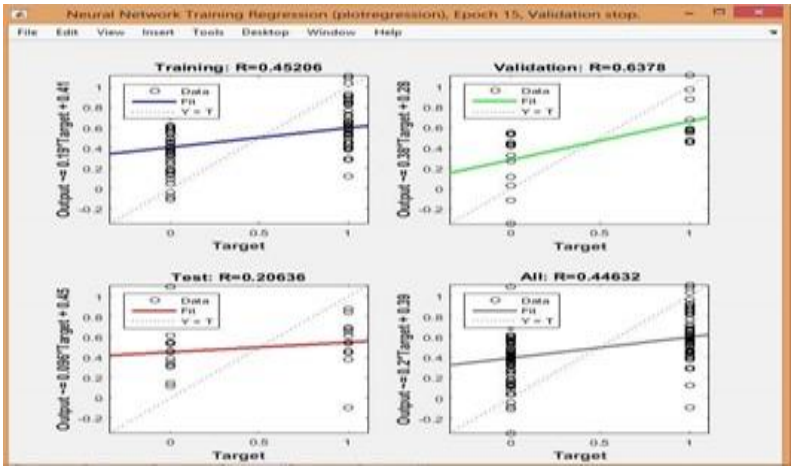

Figure 6: Regression graph after training of neural network

Regression graph shows that after training of 75% dataset for training, 15% for validation and 15% for testing, the bubbles is how close to desired output (i.e. Overall risk) in software projects.

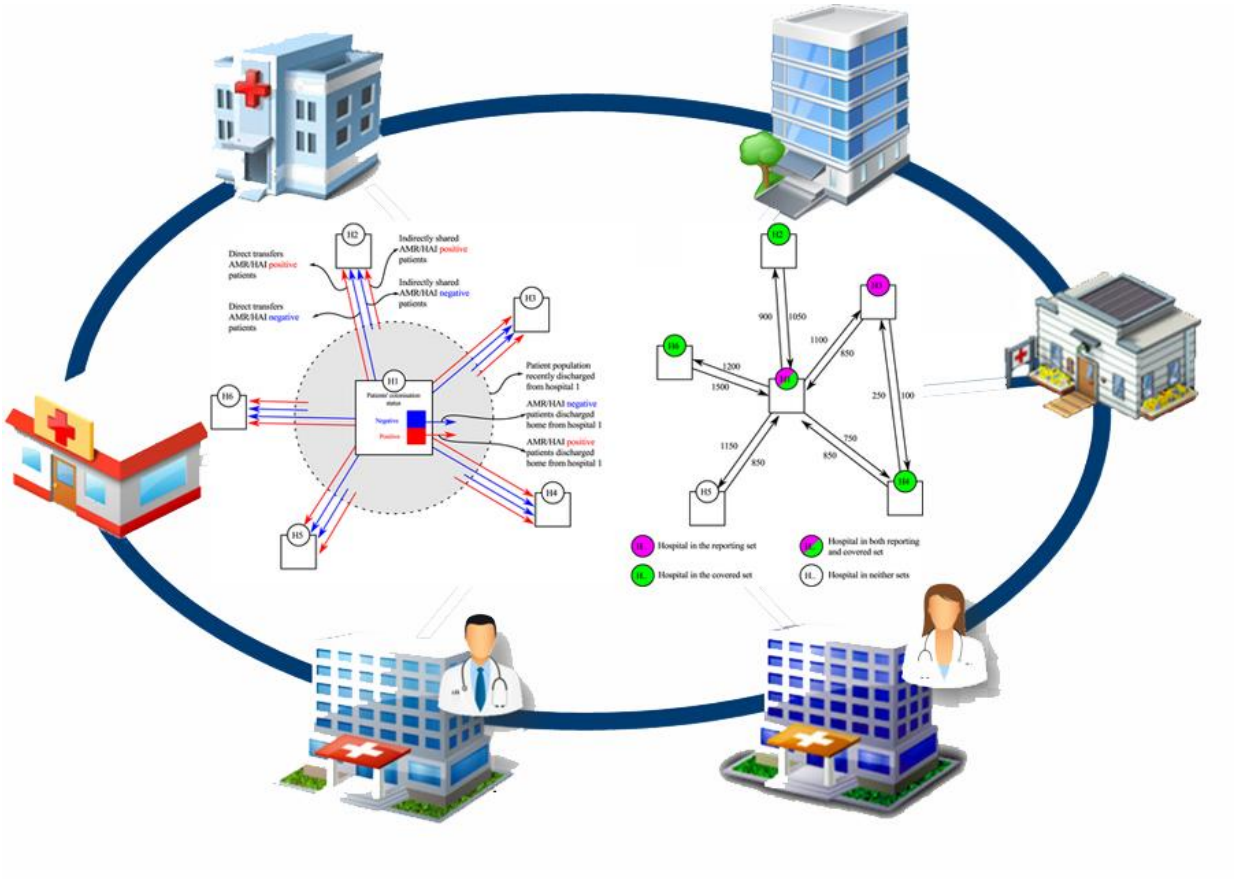

Figure 7: Optimized Data Framework for Hospital Network

Table 5: Overall risk estimation using different risk parameters in proposed model

Accuracy of different fuzzy parameters based on different risk categories for estimating risk probability			
Fuzzy Factors	**Low Contingency**	**Moderate Contingency**	**High Contingency**
Personal Risk	8.88	9.92	20.35
Process Risk	6.65	8.36	46.78
Product Risk	4.02	4.78	14.45
Platform Risk	2.41	6.37	15.48
Reuse Risk	2.17	1.54	2.32
Overall Risk Probability	94.50%	96.19%	95.74%

7. Conclusion and Future Work

The purpose of this work is to enhance the project preparation process at the beginning of the software development life cycle by way of risk management. For software programs the theoretical model is developed for the exact risk assessment. During the first step of the product development life cycle, the risk management process is developed using a neuro-fuzzy methodology. This is to the benefit of the project manager. This approach varies from the rest of the risk management process, as in software projects it uses ambiguous variables for specific risk parameters. Risk assessment and estimation of effort are two major project planning activities. Estimation of effort is mainly influenced by the probability of risk assessment at

the early stage. As comparison to the initiative and expense estimation risk evaluation, each project also has its own specific characteristics and fewer numbers of professionals are involved. The function is challenging to execute. It also adds additional cost and time to cover every new risk involved in various projects. The proposal includes a risk assessment model, which describes the following characteristics, to resolve the above problems. Precise results are obtained with fugitive techniques that take the form of language variables into the input. Vague sub parameters create certain risk chances in which precisely the fuzzy logic is measured. The overall risk volatility of the project is calculated by a qualified neural network data collection that is totally dependent on the KLOC risk category and scale. The model's learning ability is provided by the NASA'98 dataset for the neural network.

References

1. S.-J. Huang, C.-Y. Lin, N.-H. Chiu et al., "Fuzzy decision tree approach for embedding risk assessment information into software cost estimation model," Journal of Information Science and Engineering, vol. 22, no. 2, pp. 297–313, 2006.
2. B. Boehm, "Software risk management ieee computer society press," Los Alamitos CA, 1989.
3. E. E. Odzaly, D. Greer, and P. Sage, "Software risk management barriers: An empirical study," in 2009 3rd International Symposium on Empirical Software Engineering and Measurement. IEEE, 2009, pp. 418–421.
4. Ankit Kumar, Vijayakumar Varadarajan, Abhishek Kumar, Pankaj Dadheech, Surendra Singh Choudhary, V.D. Ambeth Kumar, B.K. Panigrahi, Kalyana C. Veluvolu,Black Hole Attack Detection in Vehicular Ad-Hoc Network Using Secure AODV Routing Algorithm, Microprocessors and Microsystems,2020,103352, https://doi.org/10.1016/j.micpro.2020.103352
5. C. Xu and Y. C. Shin, Intelligent Systems: Modeling, Optimization, and Control. CRC Press, 2008.
6. C. S. Reddy and R. KVSVN, "An optimal neural network model for software effort estimation," Int. J. of Software Engineering, IJSE, vol. 3, no. 1, pp. 63–78, 2010.
7. Kumar, A., & Sinha, M. (2019). Design and analysis of an improved AODV protocol for black hole and flooding attack in vehicular ad-hoc network (VANET). Journal Of Discrete Mathematical Sciences And Cryptography, 22(4), 453-463. https://doi.org/10.1080/09720529.2019.1637151
8. Kumar, A., Dadheech, P., Singh, V., Poonia, R., & Raja, L. (2019). An improved quantum key distribution protocol for verification. Journal Of Discrete Mathematical Sciences And Cryptography, 22(4), 491-498. https://doi.org/10.1080/09720529.2019.1637153
9. Dadheech, P., Goyal, D., Srivastava, S., & Kumar, A. (2018). A scalable data processing using Hadoop & MapReduce for big data. J. Adv. Res. Dyn. Control. Syst, 10, 2099-2109.
10. Kumar, A., Goyal, D., & Dadheech, P. (2018). A novel framework for performance optimization of routing protocol in VANET network. J. Adv. Res. Dyn. Control. Syst, 10, 2110-2121.
11. Kumar, A., & Sinha, M. (2019). Design and development of new framework for detection and mitigation of wormhole and black hole attacks in VANET. Journal Of Statistics And Management Systems, 22(4), 753-761. https://doi.org/10.1080/09720510.2019.1609555
12. Kumar, A., Dadheech, P., Kumari, R., & Singh, V. (2019). An enhanced energy efficient routing protocol for VANET using special cross over in genetic algorithm. Journal Of Statistics And Management Systems, 22(7), 1349-1364. https://doi.org/10.1080/09720510.2019.1618519
13. Dadheech, P., Kumar, A., Choudhary, C., Beniwal, M., Dogiwal, S., & Agarwal, B. (2019). An enhanced 4-way technique using cookies for robust authentication process in wireless network. Journal Of Statistics And Management Systems, 22(4), 773-782. https://doi.org/10.1080/09720510.2019.1609557
14. Kumar, A., Dadheech, P., Beniwal, M. K., Agarwal, B., & Patidar, P. K. (2020). A Fuzzy Logic-Based Control System for Detection and Mitigation of Blackhole Attack in Vehicular Ad Hoc Network. In Microservices in Big Data Analytics (pp. 163-178). Springer, Singapore.
15. Abhishek Kumar, et al. "The state of the art of deep learning models in medical science and their challenges". Multimedia Systems. (2020).

16. Abhishek Kumar, et al. "Efficient data transfer in edge envisioned environment using artificial intelligence based edge node algorithm". Transactions on Emerging Telecommunications Technologies. (2020).
17. Ambeth Kumar, V.D. et al. "Active volume control in smart phones based on user activity and ambient noise". Sensors (Switzerland) 20. 15(2020): 1-17.
18. Vengatesan, K. et al. "Analysis of Mirai Botnet Malware Issues and Its Prediction Methods in Internet of Things". Lecture Notes on Data Engineering and Communications Technologies 31. (2020): 120-126.
19. Vimal, V. et al. "Artificial intelligence-based novel scheme for location area planning in cellular networks". Computational Intelligence. (2020).
20. Kumar, A. et al. "Comparative Analysis of Data Mining Techniques to Predict Heart Disease for Diabetic Patients". Communications in Computer and Information Science 1244 CCIS. (2020): 507-518.
21. Sayyad, S. et al. "Digital Marketing Framework Strategies Through Big Data". Lecture Notes on Data Engineering and Communications Technologies 31. (2020): 1065-1073.
22. Kumar, V.D.A. et al. "Exploration of an innovative geometric parameter based on performance enhancement for foot print recognition". Journal of Intelligent and Fuzzy Systems 38. 2(2020): 2181-2196.
23. Vengatesan, K. et al. "Secure Data Transmission Through Steganography with Blowfish Algorithm". Lecture Notes on Data Engineering and Communications Technologies 35. (2020): 568-575.
24. Lone, T.A. et al. "Securing communication by attribute-based authentication in HetNet used for medical applications". Eurasip Journal on Wireless Communications and Networking 2020. 1(2020).
25. Vengatesan, K. et al. "Simple Task Implementation of Swarm Robotics in Underwater". Lecture Notes on Data Engineering and Communications Technologies 35. (2020): 1138-1145.
26. Kesavan, S. et al. "An investigation on adaptive HTTP media streaming Quality-of-Experience (QoE) and agility using cloud media services". International Journal of Computers and Applications. (2019)
27. Ankit Kumar, Vijayakumar Varadarajan, Abhishek Kumar, Pankaj Dadheech, Surendra Singh Choudhary, V.D. Ambeth Kumar, B.K. Panigrahi, Kalyana C. Veluvolu, Black Hole Attack Detection in Vehicular Ad-Hoc Network Using Secure AODV Routing Algorithm, Microprocessors and Microsystems, 2020, 103352, https://doi.org/10.1016/j.micpro.2020.103352