www.ijecs.in
*International Journal Of Engineering And Computer Science ISSN:2319-7242*
*Volume 4 Issue 1 January 2015, Page No. 10136-10239*

# Software Project Risk Assessment based on Neuro-Fuzzy Technique

*S. W. Jadhao[1], Prof. R. V. Mante [2], Dr. P. N. Chatur[3]*

[1] Government College of Engineering, Amravati 444 604,
Maharashtra, India
*jadhao.sudhir@gmail.com*

[2] Government College of Engineering, Amravati 444 604,
Maharashtra, India
*mante.ravi@gmail.com*

[3] Government College of Engineering, Amravati 444 604,
Maharashtra, India
*prashant.chatur@yahoo.com*

**Abstract:** ***Many risks are involved in the development of software project and Risk assessment methods are most important component in the process of risk management. They are a critical component of software project management and software testing. Practitioners, particularly researchers, are mostly interested in the evaluation of these methods for their applicability, strengths, and weaknesses for particular scenarios. So far, no model has proved to be successful at effectively and consistently predicting software development cost. Fuzzy Ex-COM (Fuzzy Expert COCOMO) that combines the advantages of a fuzzy technique with Expert COCOMO methodology for risk assessment in a software project which leverages existing knowledge and expertise from previous effort estimation activities to assess the risk in a new software project. A novel Neuro-fuzzy Expert Constructive Cost Model is proposed to improve the accuracy of risk assessment technique. With the introduction of the Neuro-Fuzzy Risk Methodology which combines the non-linear learning features of neural networks with fuzzy logic that has capability to deal with sensitive and linguistic data and generate risk rules using Artificial Neural Network(ANN) techniques to improve the accuracy of risk assessment technique. This paper shows the workflow required for implementing the Neuro-Fuzzy Risk Methodology on the original Fuzzy Ex-COCOMO methodology.***

**Keywords:** Project risk assessment, fuzzy logic, ANN, Expert COCOMO.

## 1. Introduction

Every organization faces a certain amount of risk, whether it's ensuring the health and well-being of their volunteers or proofing their products. Risk is virtually anything that threatens or limits the ability of a community or organization to achieve its mission. A number of studies have been completed that look into the success or failure rates of software projects. The latest CHAOS Summary 2009 reports that 32% of all projects were delivered on time, on budget, with required features and functions6. 44% were challenged, which were late or over budgeted, and the rest 24% failed which were canceled prior to completion. These studies indicate that serious problems exist across a broad cross-section of industries. Risk management is generally grouped with effort estimation in the software project planning process. It involves the process of identifying, approaching and mitigating the risks before any error can occur.

As compared to an effort estimation activity, which is generally done in the planning phase of the software development life cycle (SDLC), risk management, especially risk assessment, in software projects is rarely practiced and is often difficult to implement. Software risk management was introduced by Boehm and Charette et al., founders of this research area, into the area of software project managements in the late 1980s.

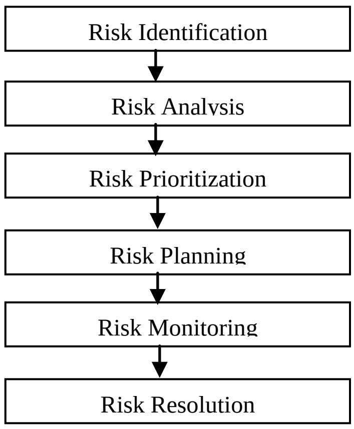



**Figure 1:** Risk Management Activities.

They classified risk into two broad categories: risk assessment and risk control. The above figure 1 shows the overall risk management activities.

Expert COCOMO is an extension of COCOMO that is used to aid in project planning by identifying, categorizing and prioritizing project risks. This research topic takes into consideration the features of neural networks and combines it with fuzzy logic, which has the capability to deal with linguistic variables, to improve the output of the project risk over Fuzzy Expert COCOMO model.

## 2. Related Work

Software Risk Evaluation (SRE) is a process for identifying, analyzing, and developing mitigation strategies for risks in a software-intensive system while it is in development. Moreover, Microsoft's research stated that an investment of merely 5% of the total budget into risk management could result in a probability of 50-70% to complete the project on time. Several Project risk model were proposed: Barry Boehm is considered among the pioneers of this field with his development of first method for risk management .Yong Hu et. al, proposed An Intelligent Model for Software Project Risk Prediction. In this paper he proposed a formal model for risk identification, and then collected actual cases from software development companies to build a risk prediction model. Hua Jiang in July 2009 has given a quantitative analysis of high-tech project risk assessment that has put risk assessment research in a good direction. The ANN to assess the high-tech project investment risk has a strong nonlinear mapping ability, with strong learning ability and high classification and prediction accuracy and RBF network is more suitable than BP network. A. V. Deursen has also given a method for risk assessment that analyses source code of a system, documentation and personnel working with or on the system. Ekananta et al. has first designed a risk model using fuzzy logic and COCOMO cost driver to calculate project risk.

## 3. Methodologies Used

In the field of artificial intelligence, Neuro-fuzzy refers to combinations of artificial neural networks and fuzzy logic. Neuro-fuzzy was proposed by J. S. R. Jang. In the proposed study, Neuro-fuzzy technique for software risk assessment which is based on the standard Expert COCOMO model, the fuzzy logic model and the neural network models. The following section briefly reviews these techniques.

### 3.1 Expert COCOMO

Expert COCOMO is an extension of COCOMO-II that is used to aid in project planning by identifying, categorizing, and prioritizing project risks. All risks in Expert COCOMO are defined as the result of a combination of several cost factors. Risk rules determine the level of every risk by mapping two cost factors (attributes) according to a risk level assignment matrix. Overall Software Project Risk quantifies the level of risk as it relates to the combination of cost factors in a software project as described in the equation formula.

Project Risk $= \sum_{j=1}^{M} \sum_{i=1}^{N} \text{Risklevel}_{ij} \times \text{Effortmultiplierprod}_{ij}$ (1)

Where

Effort multiplier product = (*driver*#1*e*ffortmultiplier) × (driver#2effortmultiplie*r*) ×...× (driver#n effort multiplier).

M= Number of Category.

N= Number of Risk Category.

The recent Expert COCOMO application was developed using a C program and an HTML interface and is posted at the USC website.

### 3.2 Fuzzy ExCOm

A fuzzy system is a mapping between linguistic terms, such as medium complexity and high cost that are attached to variables. Thus an input into a fuzzy system can be numerical or linguistic with the same applying to the output. A typical fuzzy system is made up of three major components: Fuzzier, fuzzy inference engine (fuzzy rules) and defuzzier. Fuzzy ExCOM (Fuzzy Expert COCOMO) is the software risk assessment methodology. Fuzzy logic improves the sensitivity of risk identification with Expert COCOMO and is applied to the cost factor parameters as the input for Expert COCOMO that usually describes the qualitative measurements such as very low, low, nominal, high, and very high. The fuzzier transforms the input into linguistic terms using membership functions that represent how much a given numerical value of a particular variable fits the linguistic term being considered. The fuzzy inference engine performs the mapping between the input membership functions and the output membership functions using fuzzy rules that can be obtained from expert knowledge of the relationships being modeled. a defuzzier carries out the defuzzication process to combine the output into a single label or numerical value as required.

### 3.3 Artificial Neural Network

Neural networks are physical cellular systems which can acquire, store and utilize experimental knowledge. Neural networks are typically organized in layers. Layers are made up of a number of interconnected 'nodes' which contain an 'activation function'.

Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. Most ANNs contain some form of 'learning rule' which modifies the weights of the connections according to the input patterns that it is presented with.

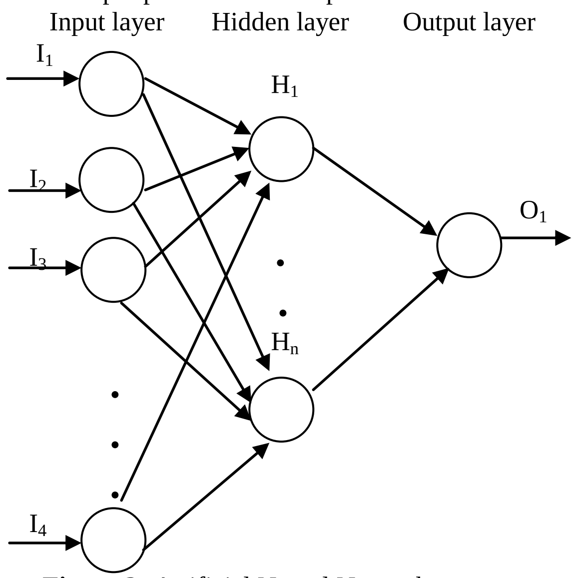



**Figure 2:** Artificial Neural Network

In a sense, ANNs learn by example as do their biological counterparts. Fig. 2 shows the generic structure of network in RBFN technique. Depending on the nature of the application and the strength of the internal data patterns one can generally expect a network to train quite well. This applies to problems where the relationships may be quite dynamic or non-linear. Neural networks provide an analytical alternative to conventional techniques which are often limited by strict assumptions of normality, linearity, variable independence etc. Because an ANN can capture many kinds of relationships it allows the user to quickly and relatively easily model phenomena which otherwise may have been very difficult or impossible to explain otherwise.

## 4. Proposed Work

To understand the effectiveness of a fuzzy system in improving risk assessment in a software project using Expert COCOMO, the proposed model is tested with COCOMO NASA93 public data set provided by PROMISE17. The data set consist of linguistic values of fifteen cost driver along with software size in KLOC and effort value in person month for each software project. The dataset has 93 project value. The block diagram in fig. 4 shows the steps involved in process of software project risk assessment.

### 4.1 Data preparation

Out of the total 93 project values first 75 project values are used for training the risk rules. The remaining 18 project values are used for testing purpose. The Expert COCOMO requires a cost factor as the input for the risk assessment activity in the COCOMO-II format. The data conversion is required for NASA93 dataset because these dataset is in the COCOMO81 data format, which is slightly different from the COCOMO-II format.

### 4.2 Data Normalization

As NASA93 dataset is available in linguistic form we have to translate data in numerical form. In this step we translated this data into their equivalent numerical value. Min-Max Normalization Performs a linear transformation on the original data. The min-max normalization is applied to this dataset to convert it into the range from zero to one. Given an attribute A, we denote by minA, maxA the minimum and maximum values of the attribute A. Max-Min Normalization maps a value v of A to v' in the range new minA, new maxA as follows.

$$v = \frac{v - minA}{maxA - minA}(\text{new minA} - \text{new maxA}) + \text{new minA} \quad (2)$$

### 4.3 Train Risk Rules

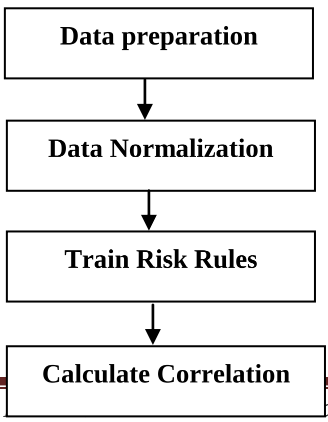
In Neuro-fuzzy system risk rules required for fuzzy system are generated using neural network.

The risk rules are evaluated using different neural network technique namely BackPropogationRBFN5.This rules are then feed to fuzzy inference engine.

.

### 4.4 Calculate Correlation

All the 105 trained risk rules are then analyzed and eliminated in this process. This analysis is done with the help of correlation coefficient. Correlation coefficient gives the relationship exists between any two attribute. It gives information about the sensitivity of project size with risk rule value



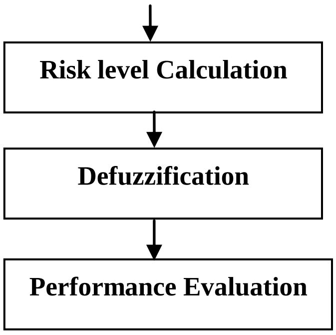




**Figure 3:** Proposed Steps Used for Risk Assessment using Neuro-Fuzzy Technique.

### 4.5 Risk Level Calculation

Using the input from fuzzification process and knowledge base (Risk Rules) generated by neural network the level of project risk is computed based on Expert COCOMO formula1.

### 4.6 Defuzzification

The Defuzzification process performs the classification of risk level value from previous step into crisp value

### 4.7 Performance Evaluation

In this study, a comparison has been made between Fuzzy ExCOM and Neuro-Fuzzy model. The performance of the model is accessed using nal MMRE and prediction accuracy (PRED) values obtained from test samples.

## 5. Conclusion

A software development project can be considered to be one of the riskier projects in the modern era. This high risk condition is driven by the uncertainty of customer requirements, the process (people, methodology, tools), and the intangible nature of the product. These factors could have a significant impact on the project schedule, the quality of the product, and the related costs. In such a situation, risk management - especially risk assessment - becomes a mandatory activity for software projects, but is often difficult and expensive to implement. Neuro-fuzzy methodology for risk assessment that combines the Fuzzy logic with the neural network to accurately classify the software project in respective risk category. We proposed model to significant 70 rules which has been chosen for evaluation of project risk while in earlier model only 31 rules derived from expertise system is used. These 70 rules is generated based on past project data not from expertise system. Future investigations in this area, which are designed to improve the accuracy and sensitivity of this methodology, can be possible by applying Genetic Algorithm (GA) Technique to find structure and parameters of neural network used.

## References

[1] B.W. Boehm, "Software risk management: principles and practices," Software, IEEE 8 (1) (1991) 32–41

[2] B.W. Boehm, T. DeMarco "Software risk management" IEEE software 14 (3) (1997) 17–19

[3] S. Chen, C. F. Cowan, P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks, Neural Networks," IEEE Transactions on 2 (2) (1991) 302–309

[4] Y. Hu, X. Zhang, X. Sun, M. Liu, J. Du, "An intelligent model for software project risk prediction, in: Information Management, Innovation Management and Industrial Engineering," 2009 International Conference on, vol. 1, IEEE, 2009.
[5] S. Mitra and Y. Hayashi, "Neuro-fuzzy rule generation: Survey in soft computing framework,," IEEE Trans. Neural Network, vol. 11, no. 3, pp. 748–767, May 2000
[6] H. Jiang, " The application of artificial neural networks in risk assessment on high-tech project investment, in: Business Intelligence and Financial Engineering," 2009. BIFE'09. International Conference on, IEEE, 2009.
[7] R. J. Madachy "Heuristic risk assessment using cost factors" IEEE software 14 (3) (1997) 51–59.
[8] A. Van Deursen, T. Kuipers, "Source-based software risk assessment, in: Software Maintenance" 2003. ICSM 2003. Proceedings. International Conference on, IEEE, 2003.
[9] J.-S. R. Jang, C.-T. Sun, "Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence", Prentice-Hall, Inc., 1996.

## Author Profile

**S. W. Jadhao** received his B.Tech. degree in Information Technology from Government College of Engineering, Amravati , Maharashtra, India in 2012, pursuing M.Tech degree in Computer Science and Engineering from Government College of Engineering, Amravati, Maharashtra, India. His research interest includes Artificial Neural Network. At present he is engaged in software project Risk Assessment which will be giving a better performance than the existing Risk Assessment model.

**Prof R. V. Mante** received his B.E degree in Computer science and Engineering from Government College of Engineering, Amravati, Maharashtra, India in 2006, the M.Tech degree in Computer Science and Engineering from Government College of Engineering, Amravati, Maharashtra, India, in 2011. He is Assistant Professor in Government College of Engineering, Amravati, Maharashtra, India, from 2007. His research interests include ECG signal analysis, soft computing technique, cloud computing. At present, He is working with Artificial Neural Network.

**Dr. P. N. Chatur** has received his M.E degree in Electronics Engineering from Government College of Engineering Amravati, Maharashtra, India and Ph.D. degree from Amravari University. He has published twenty papers in international journals. His area of research includes Artificial Neural Network, Data Mining, Data Stream Mining and Cloud Computing. Currently, he is Head of Computer Science and Engineering & Electronics Engineering Department at Government College of Engineering Amravati, Maharashtra, India. At present he is engaged with large database mining analysis and stream mining.