Numerous algorithms were exploited in risk analysis, which include Bayesian Belief Network (BBN), Artificial Neural Network (ANN), Discriminant Analysis(DA), Decision Tree (DT) \cite{hu2007software}, .

SVM

Interestingly, SVM is a powerful supervised learning method. Comparing to NN, SVM uses quadratic optimization, which theoretically should avoid the inevitable local optima problem in NN [18]. It can be hypothesized that SVM is superior to NN in performance, and some believe that it is a more promising classification method that may someday supercede NN [19]. However, can SVM maintain its superiority when the local optima problem is solved in NN, especially when complex relationships exist in data? Although SVM and NN have been employed in many applications [20,21,22], so far we have not found any established SVM model that evaluates project-scale risk.

**Software Project Risk Management Modeling with**

**Neural Network and Support Vector Machine Approaches**

Although these methods have been applied to the analysis of software quality and risks, neural network is often better than Decision Tree and Discriminant Analysis [10,14], since NN has an excellent learning and analytical abilities in complex and nonlinear problems. In contrast to the BBN approaches, NN does not require the establishment of the relations and conditional probability table, which is relatively subjective and unreliable. Neural network is an effective solution when it is difficult to build the relationships. Furthermore, neural network has other advantages in software risk measurement [10,15,16]. First of all, neural network can deal with noisy data effectively. Since the project samples are acquired through the project stakeholders’ memory, there will be a loss of information authenticity. Secondly, neural network is flexible in sample quantity. For instance, Khoshgoftaar and other researchers only collected 56 module samples to perform network training. Thirdly, neural network can deal with not only precise data but also fuzzy data. Neural network can process simultaneously quantitative and qualitative data.

@inproceedings{hu2007software,

title={Software project risk management modeling with neural network and support vector machine approaches},

author={Hu, Yong and Huang, Jiaxing and Chen, Juhua and Liu, Mei and Xie, Kang},

booktitle={Natural Computation, 2007. ICNC 2007. Third International Conference on},

volume={3},

pages={358--362},

year={2007},

organization={IEEE}

}

**Software Project Risk Assessment Model Based on Fuzzy Theory**

\cite{yu2011software} comes up with a new model of software project risk assessment, based on the fuzzy theory. It overcomes the difficulty of qualitative indicators and quantitative assessment in the traditional analysis methods. Practice shows that using fuzzy language to assess the loss of risk and adopting fuzzy logic process technology to calculate can help to solve the problems of uncertainty in the expert assessment.

@article{yu2011software,

title={Software Project Risk Assessment Model Based on Fuzzy Theory},

author={Yu, PEI},

journal={Computer Knowledge and Technology},

volume={16},

pages={049},

year={2011}

}

**Software effort estimation using Neuro-fuzzy approach**

In addition, \cite{saxena2012software} explored neuro-fuzzy techniques to design a suitable model to utilize improved estimation of software effort for NASA software projects.

@inproceedings{saxena2012software,

title={Software effort estimation using Neuro-fuzzy approach},

author={Saxena, Urvashi Rahul and Singh, SP},

booktitle={Software Engineering (CONSEG), 2012 CSI Sixth International Conference on},

pages={1--6},

year={2012},

organization={IEEE}

}

**Improving the accuracy in software effort estimation: Using artificial neural network model based on particle swarm optimization**

\cite{dan2013improving} proposed an artificial neural network (ANN) prediction model that incorporates with Constructive Cost Model (COCOMO) which is improved by applying particle swarm optimization (PSO), to provide a method which can estimate the software develop effort accurately. The result indicated that, compared with original ANN-COCOMO II model, the accuracy of software effort estimation has increased by applying PSO-ANN-COCOMO II model.

@inproceedings{dan2013improving,

title={Improving the accuracy in software effort estimation: Using artificial neural network model based on particle swarm optimization},

author={Dan, Zhang},

booktitle={Service Operations and Logistics, and Informatics (SOLI), 2013 IEEE International Conference on},

pages={180--185},

year={2013},

organization={IEEE}

}

\cite{mizuno2001prediction} have proposed a new prediction method for risky software projects, the authors used the logistic regression model to predict whether a project becomes risky or not. However, the proposed estimating approaches for the cost and the duration do not have absolutely high level of accuracy.

@inproceedings{mizuno2001prediction,

title={On prediction of cost and duration for risky software projects based on risk questionnaire},

author={Mizuno, Osamu and Adachi, Takuya and Kikuno, Tohru and Takagi, Yasunari},

booktitle={Quality Software, 2001. Proceedings. Second Asia-Pacific Conference on},

pages={120--128},

year={2001},

organization={IEEE}

}

Classification and Metaclassification in Large Scale

Data Mining Application for Estimation of Software

Projects

\cite{dzega2010classification} presented results of risk analysis experiments performed using C4.5, RandomTree and Classification and Regression Tree (CART) algorithms. Besides, they described how boosting and bagging metaclassifiers were applied to improve the results and also analyzed influence of their parameters on generalization abilities in prediction accuracy. Due to a large number of unordered labelled attributes in data sets, ANN and SVM were rejected at early stages, producing low accuracy for each dataset.

@inproceedings{dzega2010classification,

title={Classification and metaclassification in large scale data mining application for estimation of software projects},

author={Dzega, Dorota and Pietruszkiewicz, Wieslaw},

booktitle={Cybernetic Intelligent Systems (CIS), 2010 IEEE 9th International Conference on},

pages={1--6},

year={2010},

organization={IEEE}

}