A Hybrid Approach of Module Sequence Generation using Neural Network for Software Architecture

Manjot Kalsi[1], Janpreet Singh[2]

[1]M.Tech Research Scholar, Lovely Professional University, Phagwara, Punjab, India
[2]Assistant Professor, Department of CSE, Lovely Professional University, Phagwara, Punjab, India

Abstract: ***Software Engineering Principals gave a new direction to software industry [7]. With the advent of software engineering principals we are now able to check the feasibility of modules under consideration which further will lead to restrict the happening of software crisis [9]. But the selection of development of modules from the bunch of modules identified is totally dependent on the past experience of the project manager of project planner. For the detection of fault prone modules which will lead to software crisis if we are not implementing then in a desired order, various predictive models can be used based on source code metrics as input for the classifiers [13]. The identification of metric subset for the enhancement of performance for predictive objective would not only improve the model but also provides strength to the structural properties of the modules. Unbalanced datasets also itself is a kind of difficulty for building predictive modeling. Unbalanced datasets are common in empirical software engineering as a majority of the modules are not likely to be faulty [8, 28]. We propose a method of applying search based metric selection and oversampling of NASA dataset [3]. This paper is the extension of our previous review paper. In this proposed method the selection approach uses the weights of Neural Network to identify the sequence of implementation of software modules.***

Keywords: Metric subset, Predictive models, unbalanced datasets, NASA dataset, Neural Network.

1. Introduction

Many software organizations adapt the solutions provided by different software life cycle models. The refining of different steps to achieve final objective by these life cycle models helps to produce quality products as well as helps to understand in advance about the feasibility or acceptance of final work done by the organization. But the main aim of organization is to produce high quality products in a timely manner as well as cost effective manner [1]. To increase the utilization of available resources, when taking mitigating actions, such as code inspection, refactoring etc., the ability to identify potentially, referenced components would assist. Predictive models have been a focus of research in empirical studies of software systems [20]. Life cycle models fulfill the quality criteria but along with the quality criteria we have to deliver or develop the products on timely bases. So at each and every step we have to improve or enhance the criteria of selection of steps of development. Number of researchers work on this area of quantification of implementation and structural design of systems [4, 30]. Their study observes that feature selection is the process of identifying a subset of features that improves a model's specific performance [5]. Many researchers suggest to use search based optimization with testing release date planning and cost estimation [31, 32]. In this study we propose a Neural Network based algorithm as a search based feature selection strategy in order to find a subset of source code metrics that will generate an implementation sequence that enhances and simplifies a predictive model of software quality [6, 10]. To built predictive models for generating efficient sequence of modules for implementation, fit databases are required which contains instances of faulty and non-faulty modules. Preparation of balanced fit dataset is also always not possible when using industrial systems. Many modules in the fit dataset are actually imbalanced i.e. these exists a large difference between the number of fault prone modules and non faulty modules. This causes performance degradation of fault-proneness modules [15, 19].

In conventional systems, the predictive accuracy of minority class is degraded since the prediction accuracy of the majority classis dominant in satisfying the objective function of model [25, 2]. In this paper we propose sampling method to balance the fit dataset in order to increase classifier performance as a classifier we took cohesion and coupling parameters of different modules under consideration. In our previous paper we proposed this technique while doing review in this field and this paper contains the result part obtained after the implementation of proposed idea. This paper shows the performance of Artificial Neural network in accordance with the proposed system. In the following sections we discussed the objectives of over work then the methodology used by us to complete the task, and then the next section contains the information about the NASA dataset KC1used to observe the working of the proposed system. At the end we have the result part in the paper. This result part shows the performance of the proposed system with the help of parameters like Mean Absolute Error, Root Mean Square Error, and Accuracy etc.

2. Proposed Objectives

The different objectives to achieve the successful implementation are discussed below. These steps are used to find the sequence of execution of modules present in the system. This sequence will produce efficient results as per the conclusion made on the bases of parameters of comparison.

- Selection of modules under consideration from KC1 dataset.

- To find the maximum value of cohesion and coupling parameters of selected modules to find the percentage of cohesion and coupling for each selected module.
- To find the collective percentage of cohesion and coupling for a single module.
- To assign weights to the nodes of Neural Network by the values of collective percentage calculated for each module.
- To find sequence of implementation of modules on the basis of weights assigned to the Neural Network as the final output of the system.
- To find the performance measures of the modules sequence as SNR, RMSE, CPU time and accuracy.

3. Methodology

The following diagram describes the flow of methodology used in the proposed system. Some of the features of the software are selected to find the sequence of implementation of modules. In the proposed system we use coupling and cohesion as parameters to take decision about the generation of module sequence. Neural Network helps to identify the exact module to be implemented as we are considering the percentage of different parameters to take decision about which module to implement next. The judgment is done on the basis of weights assigned to the nodes of the Neural Network.

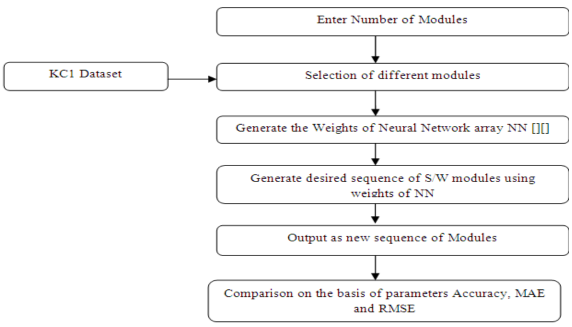

Figure 1: Proposed Methodology

4. NASA Dataset

In this study we used dataset form NASA project available from the PROMISE repository. Project KC1 is used for storage management and processing of ground data. It was developed in C++ [33, 24]. A total of twenty one metrics calculated per method for KC1, are presented in the following table:

Table1: KC1 source code metrics

loc_blank	branch_count
loc_code_and_comments	loc_comments
cyclomatic_complexity	design_complexity
essential_complexity	loc_executable
halstead_content	halstead_difficulty
halstead_effort	halstead_error_est
halstead_level	halstead_length
halstead_prog_time	halstead_volume
num_operands	num_operators
num_unique_operands	num_unique_operators
loc_total	

5. Artificial Neural Network

An Artificial Neural Network (ANN) is an information-processing paradigm that is inspired by the way a biological nervous system in human brain works [11, 18]. Large number of neurons present in the human brain forms the key element of the neural network paradigm and act as elementary processing elements. These neurons are highly interconnected and work in unison to solve complex problems. Likewise, an Artificial Neural Network can be configured to solve a number of difficult and complex problems. ANNs find a wide variety of applications in diverse areas including functional approximation, nonlinear system identification and control, pattern recognition and pattern classification , optimization , English text pronunciation, protein secondary structure prediction and speech recognition [12, 29] . Fundamentals of ANNs and its salient features are discussed in the following sections.

6. Results

In this results section we are considering the data obtained from different runs of the system by considering different number of modules each time. After considering the different modules there different values for Accuracy, MAE and RMSE considered for checking the accuracy and efficiency of proposed system. The less values of MAE and RMSE and the high values of Accuracy shows the efficient working of the system proposed with the help of Artificial Neural Network.

Table 2: Results obtained

No. of particles	Starting module	Accuracy	MAE	RMSE
10	50	97.71	12.57	15.97
20	60	97.93	12.45	16.66
30	70	98.54	12.65	16.2
40	50	97.2	11.82	14.93
50	50	97.6	13.75	16.73

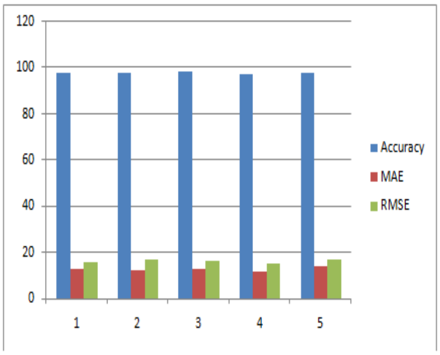

Figure 2: Chart showing results for different modules

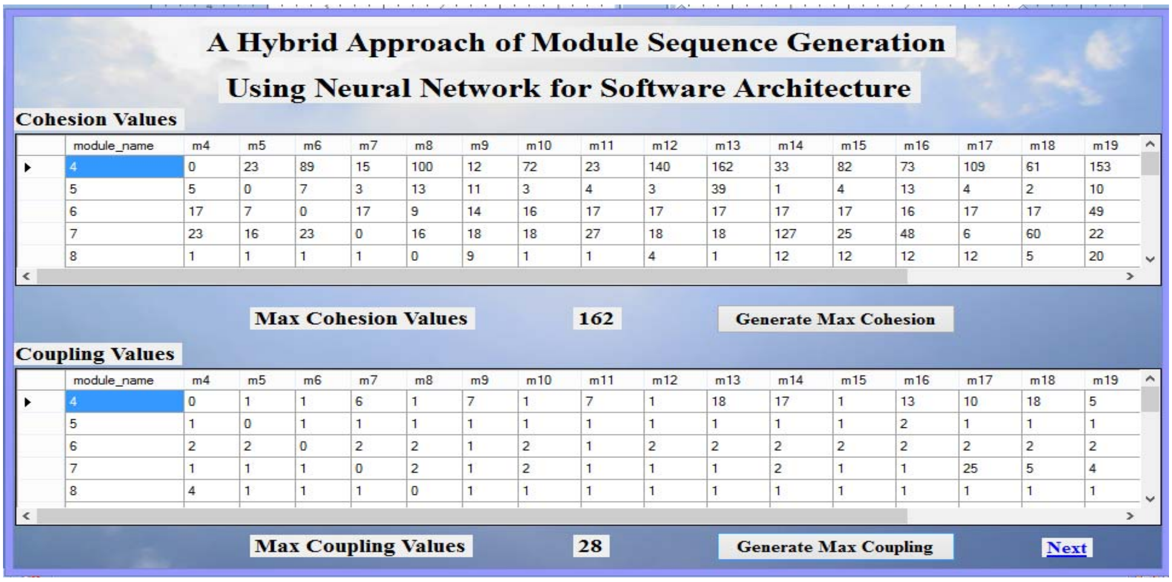

Figure 3: Uploading of Coupling and Cohesion Values

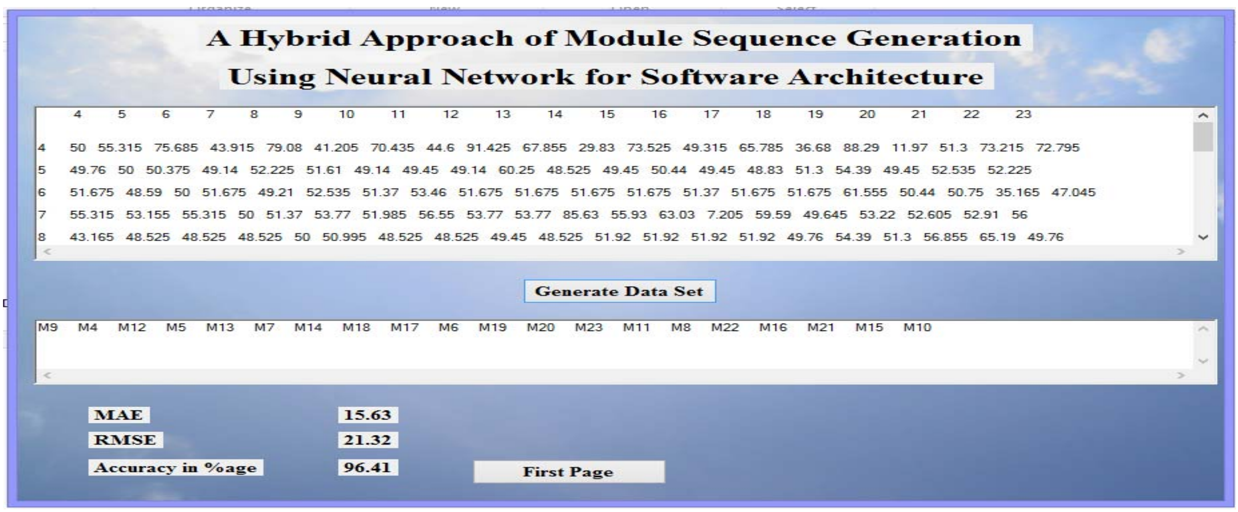

Figure 4: Final Execution Sequence generated by the System

7. Conclusion

From the literature review, it has been observed that, after deciding the architectural representation during the design phase while we are following some Model of software engineering, the choice of selecting a module from the architecture for coding and development is also very much important and we have to decide the sequence of selection of modules to go with the development. Any mistake during this selection process may leads to software crisis as if, after the implementation of some modules it is found that the section process was not good and we are not able to proceed further. So we have to start from the beginning. This may affect the overall cost of development, Time deadlines of the developers etc [14, 22]. So the selection sequence of development of modules is very important and somehow this aspect of software systems is not considered in the software implementation models. To perform the perfect selection of sequence of modules for development, we proposed a new technique with the help of Neural Networks, which helps to generate this sequence. The selection process depends on the properties of modules specified during design phase of software development cycle. These properties are coupling between the modules, Lack of Cohesion, etc. These properties will helps to decide the weights of Neural Network and then with the help of Neural Network generation of sequence of execution of modules is completed. For the performance evaluation of the proposed system, I will use KC1 data set of NASA. This data set consist 22 different properties related to 2000 different modules. SNR, RMSE, CPU time and Accuracy are the parameters under consideration.

References

[1] M. R. Lyu, Handbook of software Reliability Engineering IEEE Computer Society Press, McGraw Hill, 1996.
[2] Renu Kumar Louisiana State University Baton Rouge Suresh Rai Louisiana State University Baton Rouge Jerry L. Trahan Louisiana State University Baton Rouge Neural-Network Techniques for Software-Quality Evaluation
[3] Yan Ma, Lan Guo and Bojan Cukic, "A Statistical Framework for the Prediction of Fault-Proneness", Department of Statistics, West Virginia University, Morgantown.
[4] S. Chidamber, and C. Kemerer, "A metrics suite for object-oriented design", IEEE Transactions on Software Engineering, 20(6), 1994, pp.476-493.
[5] V. R. Basili, L. C. Briand, W. L. Mélo: A validation of object-oriented design metrics as quality indicators, IEEE Trans. on Software Eng., Vol. 20, No. 22, pp. 751-761 (1996).
[6] P.M. Khoshgoftaar, E.B. Allen, J.P. Hudepohl, and S.J.Aud, "Application of neural networks to software quality modeling of a very large telecommunications system", IEEE Transactions on Neural Network, vol. 8, pp. 902-909, 1997.
[7] T. Kamiya, Kusumoto, S., K. Inoue,"Prediction of fault proneness at early phase in object-oriented development", Proceedings of the Second IEEE International Svmposium on Object-Oriented Real-Time Distributed Computing, pp., 253- 258,1999.
[8] Mahaweerawat, A. (2004), "Fault-Prediction in object oriented software's using neural network techniques", Advanced Virtual and Intelligent Computing Center (AVIC), Department of Mathematics, Faculty of Science, Chulalongkorn University, Bangkok, Thailand, pp.1-8.Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'OZ) , Vol. 5 Lip0 W a g Jagath C. Rajapakse, Kunihiko Fukushima, Soo-Young Lee, and Xin Yao (Editors)
[9] Parvinder S. Sandhu, Satish Kumar Dhiman, Anmol Goyal World Academy of Science, Engineering and Technology 60 2009. A Genetic Algorithm Based Classification Approach for Finding Fault Prone Classes
[10] Munson, J. C. & Khoshgoftaar, T. M., "The detection of fault-prone programs", IEEE ransactions on Software Engineering, (1992), 18(5), 423-433.
[11] Mie Mie Thet Thwin Tong-Seng Quah School of Electronic & Electrical Engineering Nanyang Technological University Application of neural network for predicting software development faults using object-oriented design metrics.
[12] Khoshgoftaar, T. M., Allen, E. B., Ross, F. D., Munikoti, R., Goel, N. & Nandi, A., "Predicting fault-prone modules with case-based reasoning". ISSRE 1997, the Eighth International Symposium on Software Engineering (pp. 27-35), IEEE Computer Society (1997).
[13] Fenton, N.E. and Neil, M, "A critique of software defect prediction models", Software Engineering, IEEE Transactions on, Volume: 25 Issue: 5, Sept. - Oct. (1999), Page(s): 675 -689.
[14] Khoshgoftaar, T.M., Gao, K.and Szabo, R.M., "An Application of Zero-Inflated Poisson Regression for Software Fault Prediction", Software Reliability Engineering, ISSRE 2001. Proceedings of 12th International Symposium on, 27-30 Nov. (2001), Page(s): 66 -73.
[15] Munson, J. and Khoshgoftaar, T., "Regression Modeling of Software Quality: An Empirical Investigation", Information and Software Technology, Volume: 32 Issue: 2, (1990), Page(s): 106 - 114.
[16] Khoshgoftaar, T.M. and Munson, J.C., "Predicting Software Development Errors using Complexity Metrics", Selected Areas in Communications, IEEE Journal on, Volume: 8 Issue: 2, Feb. (1990), Page(s): 253 -261.
[17] Khoshgoftaar, T. M., Allen, E. B., Kalaichelvan, K. S. & Goel, N., "Early quality prediction: a case study in telecommunications". IEEE Software (1996), 13(1), 65-71.
[18] Khoshgoftaar, T. M. and Seliya, N., "Tree-based software quality estimation models for fault rediction", METRICS 2002, the Eighth IIIE Symposium on Software Metrics (pp. 203-214). IEEE Computer Society (2002).
[19] Seliya N., Khoshgoftaar, T.M., Zhong S., "Analyzing software quality with limited fault-proneness defect data", Ninth IEEE international Symposium on 12-14 Oct, (2005).
[20] Naeem Seliya, Member, IEEE, and Taghi M. Khoshgoftaar, Member, IEEE software Quality Analysis of Unlabeled Program Modules With Semisupervised Clustering
[21] Yogesh Singh, Arvinder Kaur, Ruchika Malhotra Software Fault Proneness Prediction Using Support Vector Machines proceedings of the World Congress on Engineering 2009 Vol I WCE 2009, July 1 - 3, 2009, London, U.K.
[22] Menzies, T., Ammar, K., Nikora, A., and Stefano, S., "How Simple is Software Defect Prediction?" Submitted to Journal of Empirical Software Engineering, October (2003).
[23] Eman, K., Benlarbi, S., Goel, N., and Rai, S., "Comparing case-based reasoning classifiers for predicting high risk software components", Systems Software, Journal of, Volume: 55 Issue: 3, Nov. (2001), Page(s): 301 – 310.
[24] Salah Bouktif, Houari Sahraoui, Giuliano Antoniol, "Simulated Annealing for Improving Software Quality Prediction", GECCO'06, July 8–12, 2006, Seattle, Washington, USA, pp. 1893-1900.
[25] Ping Guo and Michael R. Lyu, "Software Quality Prediction Using Mixture Models with EM Algorithm", Proceedings of the First Asia-Pacific

Conference on Quality Software,(APAQS 2000), Hong Kong, pp. 69-78.
[26] Qiu-suo HU and Cheng Zhong, “Model of predicting software module risk based on neural network”(in Chinese), computer Engineering and Applications, Vol.43, No.18, pp.106-110, 2007.
[27] Yan ZHAO, Cheng ZHONG, Zhi LI, Tie YAN, “Object-Oriented Software Fault-Proneness Prediction Using Support Vector Machine” (in Chinese), Computer Engineering & Science, Vol.30, No.11,pp.115-117,2008.
[28] G. Pai, "Empirical analysis of Software Fault Content and Fault Proneness Using Bayesian Methods", IEEE Transactions on software Engineering, 33(10), pp. 675-686, 2007
[29] K. O. Elish and M. O. Elish, “Predicting defect-prone software modules using support vector achines,” The Journal of Systems and Software, vol. 81, pp. 649-660, 2008.
I. Gondra, “Applying machine learning to software fault-proneness prediction,” Journal of Systems and Software, vol. 81, no. 2, pp. 186- 195, 2008.
[30] M. Shin and A. L. Goel, “Empirical data modelling in software engineering using radial basis functions,” IEEE Trans. on Software Engineering, pp. 567–576, 2000.
[31] B. Clark, S. Devnani-Chulani, and B. Boehm, “Calibrating the COCOMO II post-architecture model,” in ICSE ’98: Proceedings of the 20th international conference on Software engineering, (Washington, DC, USA), pp. 477–480, IEEE Computer Society, 1998.
[32] T. M. Khoshgoftaar, E.B. Allen, and Z. Xu, “Predicting testability of program modules using a neural network,” Proc. 3rd IEEE Symposium on Application-Specific Systems and Sof. Eng. Technology, 2000, pp. 57-6

Authors Profile

Ms. Manjot Kalsi received Msc (computer Science) degree from Khalsa College Amritsar and MCA from Kuruksherta University. She received Bsc. (Computer Science) degree from Baring Union Christian College Batala. She started her career as Assistant Professor at Baring Union Christian College Batala in department of Computer Science. She is pursuing Mtech (Computer Science & Engineering) from Lovely Professional University. She is presenting this paper as a research scholar.

Mr. Janpreet Singh received his B.Tech degree in IT from MIMIT, Malout (Punjab), INDIA, in 2002, and M.Tech degree in CSE from Guru Nanak Dev Engineering College, Ludhiana (Punjab), INDIA, in 2008. He is working as an Assistant Professor in Department of CSE at Lovely Professional University (Punjab), INDIA. His research interests include Databases and Software Engineering.