Exploration for Software Reliability using Neural Network-Based Classification method

Chitra S.[1], Madhusudhanan B.[2], Rajaram M.[3]

Abstract: Software reliability is an important aspect of software quality. According to ANSI, it is defined as "the probability of failure-free operation of a computer program in a specified environment for a specified time". One of reliability's distinguishing characteristics is that it is objective, measurable, and can be estimated, whereas much of software quality is subjective criteria. This distinction is especially important in the discipline of SQA. These measured criteria are typically called software metrics. Although software reliability is defined as a probabilistic function, and comes with the notion of time, we must note that, software reliability is different from traditional hardware reliability, and not a direct function of time. Electronic and mechanical parts may become "old" and wear out with time and usage, but software will not rust or wear-out during its life cycle. Software will not change over time unless intentionally changed or upgraded. Neural Network-based Classification Method (NNCM) was used to classify the data using recordset cyclomatic density and design density. The records were preprocessed using normal distribution. The overall error in the classification using NNCM after normal distribution was found to be 0.38%. The reliability of classification with goodness of fit measure results in and forms the subsequent improvement of error classification among the dataset.

Index terms: Neural Network, Data normalization, dataset, XLMiner, NNCM, software reliability

INTRODUCTION

Neural networks consist of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. In many applications the units of these networks apply a sigmoid function as an activation function.

The feed-forward neural network was the first and arguably simplest type of artificial neural network devised. As the majority of faults are found of its modules, there is a need to investigate the modules that are affected severely as compared to other modules and proper maintenance has to be done on time, especially for the critical applications (Ardil et al.2009). This investigates classification method based on neural network methods with data aggregation using normal density methods to test for goodness of fit and randomness.

NEURAL NETWORK CLASSIFICATION METHOD (NNCM) ARCHITECTURE

NNCM is used to classify the data with data aggregation in cyclomatic density and design density. Then these density data are normalized by using formula 1 and 2. Normalized data are trained in Neural Network Multilayer Forward Classification. Here the information moves in only one direction, forward, from the input nodes, through the hidden nodes and to the output nodes.

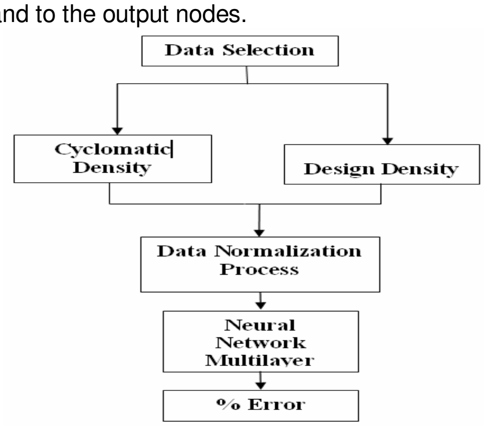

Figure 1 NNCM structure

There are no cycles or loops in the network. The improvement in percentage of classification will be obtained as result.

DATA NORMALIZATION PROCESS SEQUENCE

The flowchart given in Figure 2 describes the process of finding normal density of cyclomatic density data and design density data.

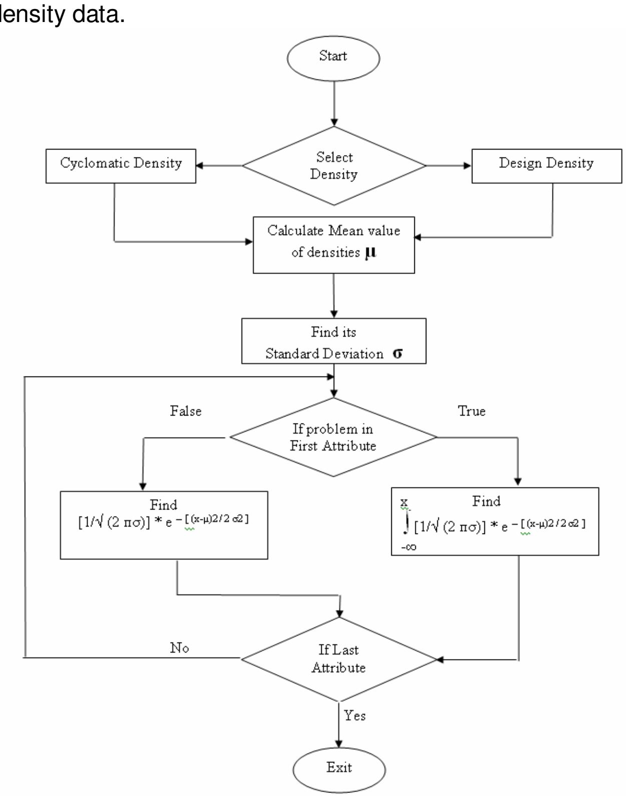

Figure 2 Data normalization process

Select Cyclomatic density and Design density to train using NNCM. For each density the following procedure is followed. First mean is calculated which is nothing but the average of all record values. Then the standard deviation is calculated using the mean value. For condition="True" the normal density is calculated using the formula.

$$f(x,\mu,\sigma)=\sum_{x=0}^{x=n}\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi\sigma}}e^{-\left[\frac{(x-\mu)^2}{2\sigma^2}\right]} \quad (1)$$

where x = Attribute value
μ = Mean of the record set
σ = Standard deviation

For condition="False" the normal density is calculated using

$$f(x,\mu,\sigma)=\sum_{x=0}^{x=n}\frac{1}{\sqrt{2\pi\sigma}}e^{-\left[\frac{(x-\mu)^2}{2\sigma^2}\right]} \quad (2)$$

The normal distribution is a continuous probability distribution whose shape is determined by the mean and standard deviation. The probability that a value is less than or equal to x will occur is the area under this curve to the left of x.
Neural network classification algorithm is applied to the prepared dataset using XLMiner. This procedure was performed for all data in the dataset.

EXPERIMENTAL INVESTIGATION OF NEURAL NETWORK CLASSIFICATION ALGORITHM

The KC3 dataset was normalized for the cyclomatic density and the design density, applying neural network classification algorithm, was computed using XLMiner and the result is given in from the Tables 1 - 6.

Table 1 Neural network classification variables

Variables		
# Input Variables	2	
Input variables	Cyclomatic Complexity Density	Design Density
Output variable	Problem	

Table 2 Neural network classification parameters

Parameters	
# Hidden layers	1
# Nodes in HiddenLayer-1	25
Cost functions	Squared error
Hidden layer sigmoid	Standard
Output layer sigmoid	Standard
# Epochs	30
Step size for gradient descent	0.1
Weight change momentum	0.6
Error tolerance	0.01
Weight decay	0

Table 3 Training data scoring - summary report

Cut off Prob.Val. for Success (Updatable)	0.5

Classification Confusion Matrix		
	Predicted Class	
Actual Class	TRUE	FALSE
TRUE	35	0
FALSE	0	140

Error Report			
Class	# Cases	# Errors	% Error
TRUE	35	0	0.00
FALSE	140	0	0.00
Overall	175	0	0.00

Table 4 Validation data scoring - summary report

Cut off Prob.Val. for Success (Updatable)	0.5

Classification Confusion Matrix		
	Predicted Class	
Actual Class	TRUE	FALSE
TRUE	39	0
FALSE	1	223

Error Report			
Class	# Cases	# Errors	% Error
TRUE	39	0	0.00
FALSE	224	1	0.45
Overall	263	1	0.38

CONCLUSION

It was seen that the error classification percentage improves if the data is preprocessed using immune technique methods. In the current research focused on the density of error rather than tupule error count. The densities were subjected to normalization and classified using neural network multilayer forward classification.
The results were promising with only 0.38% classification error. More investigation needs to be carried out to find whether the attributes selected by cyclomatic and design density are the right choice.

Table 5 Inter-layer connections weights

	Input Layer		
Hidden Layer # 1	Var1	Var2	Bias Node
Node # 1	-0.964474	-0.927539	-0.700997
Node # 2	-0.0894334	-0.0430306	-0.0495194
Node # 3	0.461368	1.12984	0.273564
Node # 4	1.40276	0.761099	0.879
Node # 5	-0.724343	-0.879881	-0.382057
Node # 6	0.462905	0.463946	0.060093
Node # 7	0.391175	0.568933	0.0368939
Node # 8	1.07879	1.05249	1.00954
Node # 9	0.162579	0.0953285	0.0410901
Node # 10	-0.704439	-0.663736	-0.149961
Node # 11	-0.806892	-1.05271	-0.693637
Node # 12	0.776518	1.01915	0.572926
Node # 13	0.875741	0.818171	0.432268
Node # 14	0.396506	0.85366	0.0480447
Node # 15	0.951899	0.950402	0.717519
Node # 16	0.611776	0.627418	0.170686
Node # 17	-0.405705	-0.356018	-.0828482
Node # 18	0.733661	0.619513	0.144205
Node # 19	0.999521	0.878044	0.646894
Node # 20	0.585231	0.637116	0.111962
Node # 21	0.404225	0.392382	0.0439063
Node # 22	0.48264	1.172	0.326712
Node # 23	0.724776	1.17979	0.663777
Node # 24	1.06117	0.962613	0.846604
Node # 25	0.298888	0.422239	-.0288435

Table 6 Hidden layer#1 output

	Output Layer	
Hidden Layer # 1	TRUE	FALSE
Node # 1	1.58043	-1.50748
Node # 2	0.174747	-0.107774
Node # 3	-1.01111	0.974191
Node # 4	-1.55759	1.5464
Node # 5	1.18307	-1.21335
Node # 6	-0.372972	0.460223
Node # 7	-0.52602	0.365629
Node # 8	-1.63846	1.54306
Node # 9	-0.122789	-0.0298756
Node # 10	1.03165	-0.900536
Node # 11	1.36712	-1.53907
Node # 12	-1.21372	1.18894
Node # 13	-1.0457	1.13599
Node # 14	-0.658415	0.696007
Node # 15	-1.29749	1.29977
Node # 16	-0.675431	0.587143
Node # 17	0.511167	-0.446454
Node # 18	-0.78179	0.697711
Node # 19	-1.35518	1.25837
Node # 20	-0.631065	0.632857
Node # 21	-0.304702	0.366621
Node # 22	-1.04512	1.08596
Node # 23	-1.31327	1.41468
Node # 24	-1.44766	1.52525
Node # 25	-0.206793	0.394103
Bias Node	0.551733	-0.597756

ABOUT THE AUTHORS

[1]Prof .S.Chitra, she is the head of the department of computer science and engineering in M. Kumarasamy College Of Engineering, Karur. She has 16 years experience in teaching. She completed her BE and ME in Computer Science And Engineering. She is doing her research in the area Software Reliability in Software Engineering. She presented more than 16 papers including national, international conferences and journals.

[2]B.Madhusudhanan , He is Student in Master of Engineering (Computer Science and Engineering), M. Kumarasamy College Of Engineering, Karur.

[3]Dr.M.Rajaram, Professor/EEE Government College of Engineering, Thirunelveli. He is guiding 12 research scholars and 4 have been awarded doctorate. He presented more than 100 papers including national, international and journals and he has 20 years experience in teaching.

REFERENCES

[1]. Ebru Ardil, Erdem Ucar and Sandhu P.S. (2009) Proceedings of World Academy of Science, Engineering and Technology, Vol. 38, No. 6,pp. 139-143.

[2]. Konstantinos Liaskos and Marc Roper (2007) Academic and Industrial Conference - Department of Computer and Information Sciences, University of Strathcl.

[3]. Dick S., Meeks A., Last M., Bunke H. and Kandel A. (2004) Fuzzy Sets and Systems, Vol. 145, No. 1, pp. 81-110.

[4]. Jing Zhao, Hong-Wei Liu, Gang Cui and Xiao-Zong Yang (2005) Proceedings of the 21st IEEE International Conference on Software Maintenance (ICSM'05)1063-6773/05 IEEE.

[5]. Andrew B. Watkins (2002), 'A resource limited artificial immune classifier', Computing Laboratory, University of Kent at Canterbury and MPI software Technology, INC. Lots C.Boggess, Intelligent Systems Laboratory, Department of Computer Science, Mississip.

[6]. Burnet, F. (1959)A Case Study for Evaluation in Science By D. R. FORSDYKE (From FASEB. Journal 1995, vol 9, 164-166 wit.

[7]. Carter, J. H. (2000) Journal of the American Medical Informatics Association 7(1).

[8]. Chotirat "Ann" Ratanamahatana Dimitrios Gunopulos ,'Scaling up the Naive Bayesian Classifier', Computer Science Department University of California Riverside, CA 92521 1-909-787-5190.

[9]. Mingxi Wu, Christopher Jermaine, 'A Bayesian Method for Guessing the Extreme Values in a Data Set', Department of Computer and Information Sciences

and Engineering, University of Florida, Gainesville, FL, USA.

[10]. Mingxi Wu, Christopher Jermaine, 'A Bayesian Method for Guessing the Extreme Values in a Data Set', Department of Computer and Information Sciences and Engineering, University of Florida, Gainesville, FL, USA.

[11]. Sommerville, I. Software Engineering, 6th edition. Addison-Wesley, 2001.

[12]. Timmis, J. and M. Neal (2001) Knowledge Based Systems 14(3/4), 121–130.

[13]. Timmis, J., M. Neal, and J. Hunt (2000) Biosystems 55(1/3), 143–150.

[14]. Tom M. Mitchell(2005), 'Generative and discriminative classifiers: naive bayes and logistic regression', Lecture notes on Machine learning.

[15]. Watkins, A. (2001), 'A Resource Limited Artificial Immune Classifier'. Master's thesis, Mississippi Sate University.

[16]. Xin Jin1, Rongfang Bie and X. Z.Gao (2006) Proceedings of the Sixth International Conference on Intelligent Systems Design.

[17]. Y.K.Malaiya and P.K.Srimani,Eds (1990), 'Software Reliability Models:Theoretical Developments,Evaluation and Applications', IEEE computer society press.