Volume 5, Issue 6, June 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Design and Development of an Algorithm for Prioritizing the Test Cases Using Neural Network as Classifier

Prof. (Dr) Amit Verma*, Simranjeet Kaur
Computer Science &Engineering
Chandigarh University
India

Abstract—Test Case Prioritization (TCP) has gained wide spread acceptance as it often results in good quality software free from defects. Due to the increase in rate of faults in software traditional techniques for prioritization results in increased cost and time. Main challenge in TCP is difficulty in manually validate the priorities of different test cases due to large size of test suites and no more emphasis are made to make the TCP process automate. The objective of this paper is to detect the priorities of different test cases using an artificial neural network which helps to predict the correct priorities with the help of back propagation algorithm. In our proposed work one such method is implemented in which priorities are assigned to different test cases based on their frequency. After assigning the priorities ANN predicts whether correct priority is assigned to every test case or not otherwise it generates the interrupt when wrong priority is assigned. In order to classify the different priority test cases classifiers are used. Proposed algorithm is very effective as it reduces the complexity with robust efficiency and makes the process automated to prioritize the test cases

Keywords—Test case prioritization, Classification, Artificial neural networks, TF-IDF.

I. INTRODUCTION

Software engineering is the study and application of engineering to the design, development and maintenance of the software. The main aim of software testing is to help the designers, developers to build a system of higher quality free from defects. Software defects are caused due to the inadequacy of software testing. Different tools, algorithms and techniques are needed for efficient and effective testing in order to complete the testing process within limited time and budget constraints. Software testing goals are defined as "The general aim of software testing is to affirm the quality of software systems by systematically exercising the software in carefully controlled experiments". Test case prioritization [15] and [3] techniques schedule test cases for execution to achieve a certain goal, such as increasing the rate of fault detection in an order that attempts to increase their effectiveness at meeting some performance goals. The goals of test case prioritization must be qualitatively stated but to measure the success of prioritization techniques in achieving these goals, however, we must define them quantitavely.

Increase the likelihood of revealing faults related to specific code changes in the regression testing process.

The field of neural networks has a history of some five decades but has found application only in the past fifteen years, and the field is still developing continuously. Neural networks are collection of elements. Neural network is able to use some hidden unknown information in the data. This process of capturing hidden information is called learning or training network. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. Various algorithms are Back Propagation algorithm [29], self organizing map algorithm. In back propagation algorithm adjusts the weights in the steepest descent direction (negative of the gradient) and calculates mean square error (MSE) that shows the difference between resulted output and target output. The Back propagation neural network used is capable of generalizing the training data. Neural network is important tool used for classification and clustering problem. An artificial neural network performs well in the classification problems in various different environments, including business, science, and engineering. Classification is most frequently encountered decision making tasks of human activity [3]. Classification is defined as when an object needs to be assigned into a predefined group or class based on numerous objects associated with object. Classification is a way of investigating the relationship between the objects to be classified and identifies the gaps in knowledge [11].For example-Bankruptcy prediction, credit scoring, medical diagnosis, quality control, handwritten character recognition, and speech recognition. Model-based testing for real-life software systems often require large number of tests and it is very time consuming and difficult task to run all the test cases so it is need of the time to prioritize the test cases[19].

This problem is solved by applying classification approach.

Many techniques and technologies have been proposed and evaluated but so none of the technology focused on to assign the priorities based on their frequency in a particular case and to check the accuracy using back propagation algorithm in artificial neural network and to differentiate between the highest and lowest priority using classifiers.

The paper's organization is as follows:

- Section 2 describes the overview of the literature.
- Section 3 describes the problem statement.

- Section 4 describes the proposed algorithm for prioritization.
- Section 5 describes the result.
- Section 6 covers the conclusion and future work.

1.1 Test Case Prioritization Techniques

Table 1: Test Case Prioritization techniques

Label	Mnemonic	Description
T1	Random	O_R
T2	optimal	O_{fd}
T3	st-total	P_{cs}
T4	st-addtl	P_{cs} not yet covered
T5	st-fep-total	P_{ef}
T6	st-fep-addtl	P_f, adjusted to consider previous test cases.
T7	fn-total	P_{cf}
T8	fn-addtl	P_{cf} not yet covered
T9	fn-fep-total	P_{ef}
T10	fn-fep-addtl	P_f previous test cases are adjusted
T11	fn-fi-total	P_{fe}
T12	fn-fi-addtl	P_{fe}, previous test cases are considered.
T13	fn-fi-fep-total	P_{fexi} & P_{fexp}
T14	fn-fi-fep-addtl	P_{fexi} & P_{fexp}, adjusted on previous coverage.
T15	fn-diff-total	P_{fe}
T16	fn-diff-addtl	P_{fe}, adjusted to consider previous test cases
T17	fn-diff-fep-total	P_{fexi} & P_{fexp}
T18	fn-diff-fep-addtl	P_{fexi} & P_{fexp}, adjusted on previous coverage

O_R - Random Order
O_{fd} - Order to optimize the fault detection
P_{cs} - Prioritize on coverage of statements.
P_{ef} - Probability of exposing faults
P_{cf} - Coverage functions
P_{fe} – Fault existence
P_{fexi},P_{fxp}-fault existence and fault exposure
P_f - Probability of faults

II. LITERATURE SURVEY

G. Rothermel, R. Untch, C.Chu, M.J.Harrold (2001**)** discussed the use of test case prioritization in regression testing and describes various techniques such as based on total coverage of code components and based on fault detection ability. The author concludes that version-specific prioritization produce improvements in fault detection rate. Investigation is done to find the alternative prioritization goals and to increase the effectiveness.

Meenakshi vanmali,Abrahem Kandel (2002**)** presented a new concept of using an artificial neural network as an automated oracle for testing software system. Author represents experimental results using a two layer neural network to detect the faults within mutated code of small credit card approval application. This approach orders the test cases into similar groups instead of ordering them according to their preference degree, so these test cases are classified using MLP neural network.

Erick Cantú-Paz, Chandrika Kamath (2005) presented a comparison of eight combinations of EAs and NNs to 8 classification problems. The author experimented with real and binary encoded EAs to train the networks. EA Algorithms used to train the network, design their architecture and select the feature subsets. Using genetic algorithms to select the feature subset yielded the most accurate classifiers. EA and NN combinations were not accurate than networks trained with simple back propagation.

Dr. Arvinder Kaur and Shivangi Goyal (2011) presented a bee colony optimization algorithm for the fault coverage regression test suite prioritization which is done by studying food foraging behavior of bees. The Effective of this algorithm has been presented using APFD metrics and chart. The challenge faced in this algorithm is the requirement of manual interface for test input data.

Shifa-e-Zehra Haidry, Tim Miller (2013) proposed a new set of techniques for functional test case prioritization based on the inherent structure of dependencies between the tests. The author concludes that test suites prioritized by this technique outperforms the random and untreated test suites but not efficient as greedy test suites.

Nida Gokce, Mubariz Eminli(2014) solved the problem by applying classification approach to the functional relationship between the test case prioritization group membership and author established the important index and frequency for all the events belonging to given group are established. The author improved the new test case model based approach in which where instead of ordering test cases according to their preference degree, they are automatically divided into groups and hence dataset is classified using MLP neural network.

III. CURRENT AREAS TO BE WORK UPON

The problem stated in this paper can be formally stated as:

There is a document having large number of words. We have to prioritize the words according to their frequency in a document and check the accuracy using back propagation algorithm. Differentiate the highest and lowest priority words with the help of classifiers for easy predictions.

Need and Significance

Various Test case prioritization techniques are implemented and discussed but yet no such algorithm in which test case prioritization is done using neural networks. Artificial neural network helps to correctly assign the priorities and it generates the interrupt when wrong priorities are assigned to different test cases. There is also a need to make the process automate as no efforts are yet made.

IV. ALGORITHM

Software testing is an important means to access the quality of software. Due to the development of 4GL which increase or speed up the implementation process, the proportion of tome devoted to testing increases. So, in order to detect the software defects or faults early, importance of test case prioritization prevails. Calculation of TF-IDF is the most important demanding part of the implemented algorithm. For Calculating the TF-IDF value , it is essential to perform tokenization,stop word removal and stemming with the help of appropriate algorithm.Optimization TF-IDF was performed using Java language, in order to calculate the frequency of a particular word in a document. TF-IDF calculates the relative frequency of each word in a document and IDF calculates in formativeness of each word over the entire document. Thus, TF-IDF value of each word is assigned as a weight in this implemented algorithm in which highest frequency word is assigned with highest priority. Different repositories are created to store the words having different priority but having same colour as used to classify them.

Algorithm_Pseudo code	Steps
Initialize T_f(T_f → Text file)	// Intialize the Pdf File
Begin	
For each D_f	
{	
(D_f= D_{tok},;D_f=D_{stem};D_f++)	
}	
Calculate D_f(TF-IDF)	// TF-IDF calculation
Assign $W_{i.....n}$ → W_d	// Assign Weights to every word
If($W_{d(freq)}$ == highest)	// highest frequency word has highest priority
{	
W_d = = H_p;	
W_d - - ;	
}	
Else	
{	
W_d = = L_p;	
W_d - - ;	
}	
Neural Network ~~Initialize~~ Weights (word, value)	
For (i=1 to n)	
{	
If ($W_{d(Freq)}$ = = $NN_{(Wi)}$)	
{	
Assign correct priority;	
}	
Else	
{	
Back Propagation Error (Re – initialize weights (word, value)	// Back propagation to generate the interrupt
}	
}	
}	
Assign $C_{(Red)}$ → $Wd_{(Highest)}$	// Classification of highest and lowest priority words by using classifier
Assign (Green) → $W_{d(Lowest)}$	

V. RESULTS AND DISCUSSION

Fig 5a: TF-IDF

Fig 5b: Back propagation Algorithm

Fig 5c: Classification of Different Priorities

Figure 5d: Highlight the priorities

Figure 5e: APFD Metric

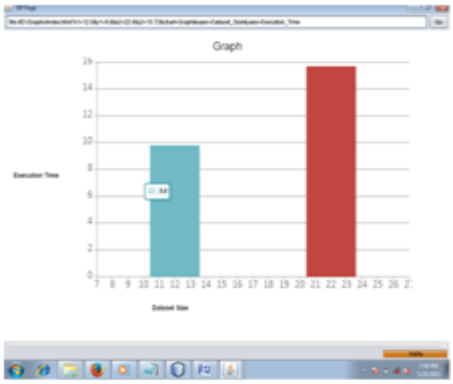

Figure 5f: Execution Time

Figure5a: After uploading the Pdf file, Tokenization, stop word removal and stemming is done to calculate term frequency of every word in the document in order to assign the weight to each word for prioritization meansTF-IDF of file is calculated.

Figure5b: Apply Back Propagation algorithm to check whether correct weights are assigned or not. In case priority is different from the actual priority of the word, neural networks back propagates the error to increase the accuracy.

Figure 5c: After assigning priorities to different words, repositories are created to store higher and lower priority test cases for easy prediction.

Figure 5d: After assigning the priorities, highlight them in the PDF file uploaded.

Figure 5e: The proposed technique has higher APFD metric as compared to existing technique.

Figure 5f: The proposed technique has very less execution time.

VI. CONCLUSION AND FUTURE SCOPE

In this paper we represent a framework to prioritize the test cases based on TF-IDF, Artificial neural networks back propagation algorithm and naïve bayes classifier algorithm. The framework with embedded methods gave good results, confirmed our concepts and initial expectations. Evaluation of the algorithm is done on various documents. The framework was very stable and reliable. Performed tests have detected a sensitivity of the implemented algorithm. The analysis of the documents content showed that the amount of unusable words in documents has significant impact on the classification so it is necessary to improve the preprocessing of documents to achieve the better results. Classification is very important to easily detect the highest priority and lowest priority word in a document. The implemented algorithm increases the robustness, efficiency and reduces the complexity.

REFERENCES

[1] Bing Cheng and DM. Titterington , " Neural Networks: A Review from Statistical Perspective", Statistical science , Vol.9,No,1, 2-54,1994. Ken-ichi Funahashi, "Multilayer neural networks and Bayes decision theory", Elsevier Science,pageno.209-213,Received 9 November 1994; accepted 22 September 1997.

[2] Guoqiang Peter Zhang, "Neural Networks for Classification: A Survey" IEEE Trans.On Man, Systems, and Cybernetics, vol.30, no.4, Nov 2000

[3] G. Rothermel, R. Untch, C.Chu, M.J.Harrold, "Prioritizing Test Cases for Regression Testing," IEEE Trans. Software Eng.,vol.27,no.10,pp.929-948,oct.2001.

[4] Meenakshi Vanmali, Mark Last, Abraham Kandel, "Using a Neural Network in the Software Testing Process", International Journal of Intelligent Systems, vol.17, page no. 45-62, 2002.

[5] Sebastian Elbaum, Alexey G. Malishevsky, G.Rothermel, "Test Case Prioritization: A Family of Empirical Studies", IEEE Trans. Software Eng., vol.28, no.2, Feb.2002.

[6] Isabelle guyon, jason weston, stephen barnhill, "Gene Selection for cancer classification using support vector machine" , Machine Learning, 46, 389–422, 2002 Kluwer Academic Publishers. Manufactured in The Netherlands.

[7] James A. Jones and Mary Jean Harrold, "Test-Suite Reduction and Prioritization for Modified Condition/Decision Coverage", IEEE transactions on software engineering, vol. 29, no. 3,page no.195-209, march 2003.

[8] Li Baoli, Yu Shiwen, and Lu Qin, "An Improved k-Nearest Neighbor Algorithm for Text Categorization" china 2003.

[9] D. Richard Kuhn, Senior Member, Dolores R. Wallace(2004), "Software Fault Interactions and Implications for Software Testing, IEEE Trans. on software engineering, vol. 30, no. 6, June 2004.

[10] 10. Erick Cantu-Paz, Chandrika Kamath, "An Empirical Comparison of Combinations of Evolutionary Algorithms and Neural Networks for Classification Problems", IEEE Trans. On Systems, Man, And Cybernetics, 2005.

[11] Zheng Li, Mark Harman, and Robert M. Hierons, "Search Algorithms for Regression Test Case Prioritization", IEEE Trans. Software Eng., vol.33, no.4, April 2007.

[12] Randall S. Sexton and Robert E.Dorsey, "Reliable Classification Using Neural Networks: A Genetic Algorithm and Back propagation Comparison".

[13] Usha Badhera, G.N Purohit, Debarupa Biswas , "Test case prioritization algorithm based upon modified code coverage in regression testing" International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.6,page no.29-37, November 2012.

[14] Stefan Wallin and Leif Land´en, "Telecom Alarm Prioritization using Neural Networks", 22nd International Conference on Advanced Information Networking and Applications – Workshops, page no.1468-1473,2008.

[15] 16. Adam Slowik, and Michal Bialko, "Training of Artificial Neural Networks Using Differential Evolution Algorithm" , IEEE Krakow, Poland, page no. 60-65,May 25-27, 2008.

[16] Shi Yu∗, Steven Van Vooren, Leon-Charles Tranchevent, Bart De Moor and Yves Moreau, "Comparison of vocabularies, representations and ranking algorithms for gene prioritization by text mining", Vol. 24 ECCB ,, pages 119–125,2008.

[17] Usman Farooq, C. P. Lam, "Evolving the Quality of Model Based Test Suite", IEEE International conference on Software Testing verification and validation workshops, 2009.

[18] Sira Vegas, Natalia Juristo, and Victor R. Basili , "Maturing Software Engineering Knowledge through Classifications:A Case Study on Unit Testing Techniques", IEEE transactions on software engineering, vol. 35, no. 4,page no.551 – 565, july/august 2009.

[19] N. Suguna1, and Dr. K. Thanushkodi, "An Improved k-Nearest Neighbor Classification Using Genetic Algorithm" IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 4, No 2, July 2010.

[20] Ajitha, Dr. T.V. Suresh Kumar, D. Evangelin Geetha, Dr. K. Rajnikanth, "Performance Prediction In Early Stages Of Software Systems: Artificial Neural Network Model" ICCCCT ,pageno.743-747,2010.

[21] "Improving Testing Efficiency:Agile Test Case Prioritization" Software Benchmarking Organization,pageno. 1-7, 2009 – 2010.

[22] Amit Ganatra, Y P Kosta Gaurang Panchal, Chintan Gajjar. "Initial Classification Through Back PropagationIn a Neural Network Following OptimizationThrough GA to Evaluate the Fitness of an Algorithm", International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, page no. 98-116,Feb 2011.

[23] Ashwin G. Raiyani and Sheetal S. Pandya, "Proritization technique for minimizing number of test cases" International Journal of Software Engineering Research & Practices Vol.1, Issue 1, page no-3-9,Jan, 2011.

[24] Arvinder kaur and Shubhra Goyal, "A genetic algorithm for regression test case prioritization using code coverage ", International Journal on Computer Science and Engineering, Vol. 3, No. 5, page no.1839-1847,May 2011.

[25] Sonali Khandai, Arup Abhinna Acharya and Durga Prasad Mohapatra, "Prioritizing Test Cases Using Business Criticality Test Value", International Journal of Advanced Computer Science and Applications,Vol. 3, No. 5, page no.103-110,2011.

[26] Zhen-Guo Che, Tzu-An Chiang and Zhen-Hua Che3, "Feed-forward neural networks training: a comparison between genetic algorithm and

[27] Back-propagation learning algorithm", International Journal of Innovative Computing, Information and Control Volume 7, Number 10, pp. 5839-5850,October 2011.

[28] Dr. Arvinder Kaur and Shivangi Goyal "A Bee Colony Optimization Algorithm for Fault Coverage Based Regression Test Suite Prioritization" International Journal of Advanced Science and Technology Vol. 29, April, 2011.

