International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 11 (2017) pp. 2879-2886
© Research India Publications. http://www.ripublication.com

Defect Classification using Relational Association Rule Mining Based on Fuzzy Classifier along with Modified Artificial Bee Colony Algorithm

[1]B. Dhanalaxmi
[1]Associate Professor, Department of Information Technology
Institute of Aeronautical Engineering, Hyderabad, India.

[2]Dr.G. Apparao Naidu
[2]Professor, Department of Computer Science and Engineering,
JB Institute of Engineering and Technology, Moinabad, Hyderabad, India.

[3]Dr.K. Anuradha
[3]Professor & Vice-Principal
School of Computing, Gokaraju Rangaraju Institute of Engineering and Technology,
Hyderabad, Pin: 500090, India.

Abstract

In this study, we introduce a method for defect classification using relational association rule mining based on fuzzy classifier along with modified artificial bee colony algorithm. Relational association rules are an extension of ordinal association rules, which are a particular type of association rules that describe numerical orderings between attributes that commonly occur in the data. These relationships may express quantitative information that may exist in the vector characterizing a software entity. Initially the features are selected from the database in preprocessing phase. The features represent software metrics extracted from the source code. In the database some features are redundant and/or irrelevant which calls for the removal. After that, the rules are generated from the selected feature subset by employing Relational Association Rule Mining (RARM). Then the generated rules are optimized with the help of Modified Artificial Bee Colony (MABC) algorithm. These optimal rules are given as the input for the classifier to identify the defects in the database. The classification can be done by fuzzy classifier. The proposed methodology will be implemented in JAVA by using Promise datasets repository.

Keywords: relational association rule mining, defect classification, fuzzy classifier, and artificial bee colony algorithm.

INTRODUCTION

With the development of computer technology, software systems became more in number and more complicated and due to limited human ability, there are a lot of defects are caused in software's life cycle development. Despite precise planning, acceptable documents and control process along with software development, specific defects are inevitable. A software defect is an error, defect, problem, mistake or failure in a computer program. This way, the system may cause a wrong or unexpected result. In face the precision of the errors that occur within codes is of importance because it can directly contribute to decreasing costs and developing software quality. Studies have shown that defects are found in only some software modules. These defects can lead to software failure, decreased customer satisfaction or increased maintenance costs [1]. Defective software components have devastating consequences on increased development and maintenance costs and declining customer satisfaction [6], [10]. Defect prediction has both safety and economic benefits in technical systems by preventing future failures and further improves process maintenance schedules. Lack of adequate tools to evaluate and estimate the cost for a software system failure is one of the main challenges in systems engineering. For a software development project, it is highly desirable to reduce software defects [4].Data mining can be a useful method in initializing the rules for defect identification in the software development.

Data can be in any form like facts, text or numeric which can be computed by computer. This data when converted to meaningful data becomes information. Knowledge is nothing but useful information. Data mining is process of finding knowledge from huge collection of data. It has various methods to extract hidden knowledge from large data set. Techniques like association rule mining, classification and clustering can be used to analyze data. Classification is the process of finding a set of models that describe and distinguish data classes and concepts, for the purpose of being able to use the model to predict the class whose label is unknown [3]. Data Mining is highly multidisciplinary field, which has its roots in statistics, mathematics, information theory, artificial intelligence, machine learning theory, data bases and in the whole series of other related fields. DM involves activities of searching large databases and data warehouses with the aim to find the hidden, so far unknown facts, regularities or patterns. Data mining represents finding useful patterns or trends through large amounts of data [9]. With the data mining techniques more mature and widely used, to analysis and mining the hidden information in software development repository become a hot research topic.

The usual ways which use data mining techniques in this domain include Association Rules, Classification and Prediction, Clustering. Classification means to build defects prediction model by learning the already existed defects data to predict the defects in future version of software [5].

There are two main stages in classification using ARM. The first one is the stage of association rule discovery or rule generation, and the other one is the stage of association rule selection or rule ranking. As a rule-based method, the ARM approaches face the challenges on rule pruning in both stages in order to reduce the time and space cost in the mining process. To address these challenges, effective interestingness measures, one of the core elements in generating rules and selecting rules (properly pruning the rules), can be utilized. In the rule generation stage, the interestingness measures are used to discover the feature-value pairs and remove the un-interesting feature-value pairs [2].

Association rule mining means searching attribute–value conditions that occur frequently together in a dataset. Ordinal association rules are a particular type of association rules. Given a set of records described by a set of attributes, the ordinal association rules specify ordinal relationships between record attributes that hold for a certain percentage of the records. However, in real world datasets, attributes with different domains and relationships between them, other than ordinal, do actually exist. In such situations, ordinal association rules are not strong enough to describe data regularities. Consequently, relational association rules were introduced in order to be able to capture various kinds of relationships between record attributes [7].

Neural networks are adept at modeling nonlinear functional relationships that are difficult to model when exploiting other techniques, and thus, form an attractive alternative for software quality modeling. However, traditional machine learning approaches may be faced with a difficulty of building a classification model (classifier) with binary class imbalanced data in software quality engineering. The class imbalanced dataset has a class that outnumbers greatly the other classes. One of the commonly encountered techniques to alleviate the problems associated with class imbalance is data sampling that can be accomplished through under sampling, oversampling or both of them [8].

RELATED WORK

Numerous researches where done in the field of defect classification in software development using different data mining rules. The increased demand in software has evenly increased the necessity to develop better defect free software which made the research a hot topic in recent years. Some of the researches made in the field of software defect classification is described below,

Software bug repository was the main resource for fault prone modules. Different data mining algorithms are used to extract fault prone modules from these repositories. Software development team tries to increase the software quality by decreasing the number of defects as much as possible. Naheed Azeem and Shazia Usmani[11] have proposed different data mining techniques for identifying fault prone modules as well as compare the data mining algorithms to find out the best algorithm for defect prediction.

Data mining is the process of extracting the previously unknown patterns from large amount of data. Privacy preserving data mining was one of the research areas in data mining. The main objective of privacy preserving data mining was to provide the privacy for personally identifiable information in the datasets. Some of them are Privacy preserving Association Rule Mining, Privacy Preserving Clustering, Privacy Preserving Classification, Statistical disclosure control, K-anonymity etc. S.Vijayarani and M.Sathiya Prabha [12] have discussed about the association rule hiding problem. Association rule mining was one of the very important data mining techniques. The process of discovering item sets that frequently co-occur in a transactional database so as to produce significant association rules that hold for the data was known as Association rule mining. Association rule hiding was the process of modifying the original database by hiding the sensitive data to protect the sensitive association rules. Here they have proposed an Artificial Bee Colony optimization algorithm for hiding the sensitive association rules. They analyzed the efficiency of the Artificial Bee Colony optimization technique by using various performance factors.

Software defect prediction plays an important role in improving software quality and it help to reducing time and cost for software testing. Machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data. The ability of a machine to improve its performance based on previous results. Pooja Paramshetti and D. A. Phalke [13] have used this survey to understand the existing techniques for predicting software defect. Machine learning improves efficiency of human learning, discover new things or structure that is unknown to humans and find important information in a document. For that purpose, different machine learning techniques are used to remove the unnecessary, erroneous data from the dataset. Software defect prediction is seen as a highly important ability when planning a software project and much greater effort is needed to solve this complex problem using a software metrics and defect dataset. Metrics are the relationship between the numerical value and it applied on the software therefore it is used for predicting defect.

Despite careful planning, clear documentation and proper software process control, software defects are inevitable. Defect causal analysis was the nucleus of process improvement approaches. Causal analysis was regarded as one of the core steps in software defect prediction. The analysis of software faults improves the reliability and quality of software products thereby adding value to the business. Many data mining techniques have been suggested for defect prediction and prevention. Remya Kartha K and Vikraman Nair R [14] have summarized the various data mining techniques used in defect prediction and defect prevention and also propose the development of a data mining based tool to automate the root cause analysis of software defects. The proposed tool makes use of Hadoop File System to store the defect data. Implementation of decision tree algorithm in Map Reduce was proposed for the classification and analysis of defects.

Software defect prediction tries to automatically identify defective software modules, in order to help software testers focus their time and effort on those modules which are likely to contain faults. So far many different machine learning algorithms have been used for this classification task. Zsuzsanna Marian [15] have introduced a software defect prediction method, called DPRAR, which uses relational association rules to classify modules, represented by a vector of software metric values, as faulty or non-faulty. Here they investigated how different feature elimination techniques influence the results of the DPRAR method. They also consider two methods for computing the scores for a module, which two values are showing how close the module was to the faulty instances and the non-faulty instances.

In recent years, Association Rule Discovery has become a core topic in Data Mining. It attracts more attention because of its wide applicability. Association rule mining was normally performed in generation of frequent item sets and rule generation in which many researchers presented several efficient algorithms. T. Karthikeyan and N. Ravikumar [16] have proposed a theoretical survey on some of the existing algorithms. The concepts behind association rules are provided at the beginning followed by an overview to some of the previous research works done on this area. The advantages and limitations are discussed and concluded with an inference.

The main aim of software development to develop high quality software and high quality software was developed using enormous amount of software engineering data which was studied by Safia Yasmeen [17]. The software engineering data could be used to gain empirically based understanding of software development. The meaning full information could be extracted using various data mining techniques. As Data Mining for Secure Software Engineering improves software productivity and quality, software engineers are increasingly applying data mining algorithms to various software engineering tasks. However mining software engineering data poses several challenges, requiring various algorithms to effectively mine sequences, graphs and text from such data. Software engineering data includes code bases, execution traces, historical code changes, mailing lists and bug data bases. They contain a wealth of information about a projects status, progress and evolution. Using well established data mining techniques, practitioners and researchers could explore the potential of this valuable data in order to better manage their projects and do produce higher-quality software systems that are delivered on time and within budget.

PROPOSED METHOD

The features are chosen from the database in preprocessing phase in the suggested methodology. Now the features symbolize the software metrics. The rules are produced by Relational Association Rule Mining (RARM) based on these metrics, which is the addition of ordinary association rule mining. The non functional requirements are estimated along with these rules. Then the rules are optimized with the help of Modified Artificial Bee Colony (MABC) algorithm. These optimal rules are specified as the input for the classifier to recognize the imperfections in the database. The classification is prepared by fuzzy classifier. By means of fuzzy classifier the database is categorized whether it is imperfections or not. The architecture diagram of the suggested defect classification is exposed in Fig. 1.

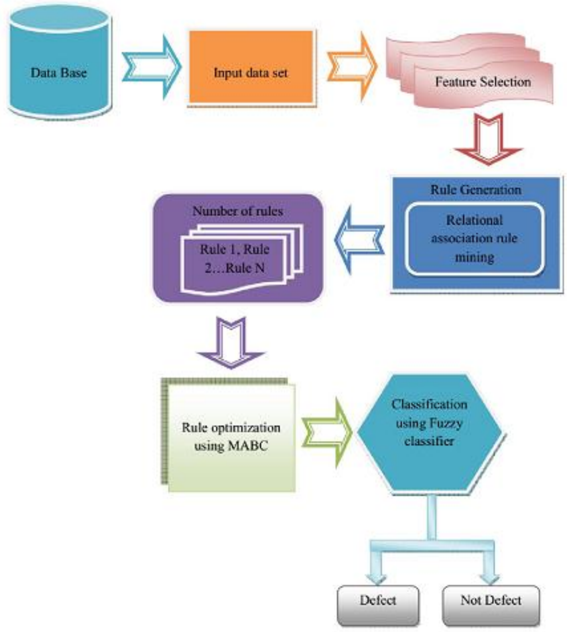

Figure 1. Architecture diagram of proposed defect classification

The proposed method use JM1 dataset for defect prediction. The software metrics used from the datasets are,

- Essential complexity
- Design complexity
- Total operators + operands
- Intelligence
- False, True

A. Rule generation by RARM (Relational Association Rule Mining)

The rule will be produced based on the chosen features or metrics. Relational association rule mining is employed to produce the rule for the suggested method. Relational association rule mining is a technique, which is the addition of ordinary association rule mining. Let as consider records in the relational model $R_{rm}= \{r_1, r_2….r_N\}$, where each record consist of M attributes $(x_1, x_2….x_M)$. The length of a relational association rule can be at most equal to the number of the attributes explaining the data. The relational association rule mining can be described relations such as, less or equal (≤), equal (=), greater or equal (≥).

With some example the specified process of relational association rule mining is made cleared. Reflect on one sample dataset, it's revealed in Table.1, we are assuming A, B, C, D are the software metrics. Rule will be produced based on this metrics.

Table 1. Sample dataset

A	B	C	D
1	0	2	1
0	3	1	0
1	0	0	2
1	0	1	0

The rule will be produced based on the relation of each feature after the feature selection. Now the length of the rule is equal to the number of characteristics in the sample dataset. A few of the rules are specified beneath,

Rule 1: A>B

Rule 2: A<C

Rule 3: A=D

Rule 4: A>B<C

Rule 5: A=D>B

Rule 6: A=D>B<C

Finally these rules are optimized to get the best rule. The proposed method use modified artificial bee colony algorithm for rule optimization. The step by step procedure of modified artificial bee colony algorithm is given below,

B. Rule Optimization using MABC

The ABC contains mainly three types of bees namely: employer bees, onlooker bees and the scot bee. At first the user described size of employee bees is arbitrarily initialized. The fitness is calculated for each employee bee. Using fitness of the employer bee the probabilities of each of the employer bee are calculated, which is turn, is employed to initialize the onlooker bee. The fitness of each of onlooker bee is calculated. If the concert of bees is not illustrating any markable development for user specified time, then the scout bee is thrown out and restored by the arbitrarily initialized employee bee. This process is replicated for user specified number of iterations. The best bee symbolizes the optimal rules. Here the traditional artificial bee colony algorithm is modified with the help of cross over and mutation process. Instead of scout bee operation the proposed method use cross over and mutation process. Here the solution is represent the rules. The detailed process is shown in Fig.2,

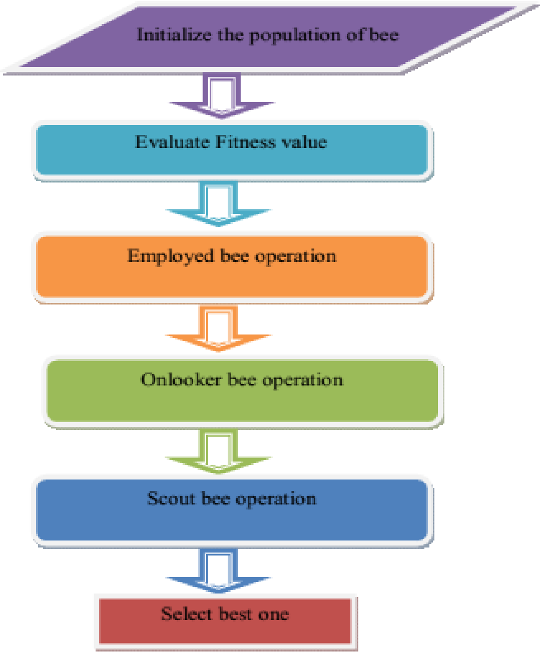

Figure 2: The process of Modified Artificial Bee Colony algorithm

Procedure of the MABC algorithm

- Initialize the population of solution Ri
- Estimate the population
- Set cycle to 1
- Do again
- Produce novel solution NR_i in the neighborhood of Ri for employed bees and relate the greedy selection process Ri and NRi.

$$NR_i = R_{ij} + \varphi(R_{ij} - R_{kj}) \qquad (1)$$

Where R_{ij} is the jth parameter of the ith employed bee; NR_i is a new solution for R_{ij} in the jth dimension; R_{kj} is the neighbor particle of R_{ij} in employed bee population; φ is a number arbitrarily chosen in the range of [-1, 1].

- Evaluate the fitness function. Here fitness function represents the objective function of the proposed method.

$$Fitness = \max accuracy \qquad (2)$$

- Calculate the probability values for the solution Ri be means of their fitness values using the equation (2).

$$\Pr obability = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_n} \qquad (3)$$

- Generate the novel solution NRi for the onlooker from the solution Ri selected depending on chance and assess them and use the greedy selection process
- If there is an thrown out solution for scout then substitute it with a novel solution by means of equation (1)
- Commit to memory the best solution so far
- Cycle=cycle+1
- Until cycle=Max iterations

At last the optimized rule is employed to fuzzy classifier for the defect classification. The general steps for fuzzy classifier is specifically described beneath,

C. Defect Classification using fuzzy classifier

Defect classification is prepared by fuzzy classifier in our suggested method. Now all the optimal rules are separated into two for the training and testing purpose. Initially some rules are trained after that the unfamiliar rules are categorized in the testing phase. Proposing of fuzzy system has three significant steps fuzzification, fuzzy inference engine and defuzzification. Fuzzification is necessary as a degree of membership function is specified for each member of set. The fuzzy system forecasts the results more precisely with the membership function. The fuzzy membership function definition and fuzzy rule base are the two significant steps. Fuzzy Membership function: The membership function is proposed by selecting the proper membership function. The proposed method selects the triangular membership function to modify over the information into the fuzzified value. The Triangular membership function contains three vertices l, m and n of in a fuzzy set Q (l: lower limit and n: upper limit where membership degree is zero, m: the centre where membership degree is one).

The formula used to compute the membership values is depicted as below,

$$f(R)=\begin{Bmatrix} 0 & if\ R\le l \\ \dfrac{R-l}{m-l} & if\ \ l\le R\le m \\ \dfrac{n-R}{n-m} & if\ \ m\le R\le n \\ 0 & if\ R\ge n \end{Bmatrix} \qquad (4)$$

For a single fuzzy set, figure 3 shows a triangular membership function. Now, we can see that at l and n the value is zero and it attains constantly to a maximum of value one at the centre point m between the l and n. The triangular membership function demonstrated in beneath,

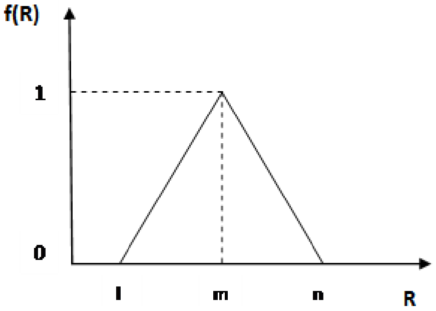

Figure 3: Triangular membership function

Rule Base: We previously created the fuzzy rule set by means of relational rule mining that is specified in the fuzzy rule base. Using the fuzzy system the defect classification is performed. The testing data with decreased attribute is specified to the fuzzy logic system, where the test data is changed to the fuzzified value based on the fuzzy membership function. Next, based on the membership function, the fuzzified input is matched with the fuzzy rules defined in the rule base. After that the output is specified to the defuzzification, now the fuzzified value is transformed to crisp value then the calculation is prepared. It is employed to categorize and predict the class for all the rules. The rules were categorized from fuzzy classifier by recognizing the defuzzification value above the threshold. If the value is above the threshold denotes the rule classified to defect class or else not defect class.

RESULTS AND DISCUSSION

The proposed Defect classification system using fuzzy classifier with modified artificial bee colony algorithm is implemented in the working platform of NETBEANS version 7.2 (jdk 1.7).The proposed method is carried out for the database JM1.The feature selection is done first and for these process the corresponding rules are generated using relational association rule mining. The number of rules is optimized by incorporating the Modified Artificial bee colony algorithm. Then the optimal rules are fed to fuzzy classifier to predict the defects. The final classified output provides the defect that exists in the system. The proposed method is evaluated using various evaluation metrics which are explained in the below section,

A. Performance Evaluation

The performance evaluation of the proposed methodology is calculated by measuring the defect density, accuracy, sensitivity and specificity of the method. The sensitivity, specificity and accuracy values are calculated using the expression given below,

$$\text{Defect density}=\frac{\text{Total number of defects}}{\text{Lines of code}} \qquad (5)$$

$$Sensitivity\ = (\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})) \qquad (6)$$

$$Specificity\ = (\mathrm{TN}/(\mathrm{FP}+\mathrm{TN})) \qquad (7)$$

$$Accuracy = 100-((FAR+FRR)/2) \qquad (8)$$

Where,

True positive (TP) is the number of defects that are correctly classified.

True negative (TN) is the number of irrelevant defects that are correctly classified.

False positive (FP) is the number of relevant defects that are incorrectly classified as irrelevant defects

$$FAR = 1-Specificity$$

False negative (FN) is the number of irrelevant defects that are incorrectly classified as relevant defects

$$FRR = 1-Sensitivity$$

The values for each of these evaluation metrics are evaluated and the values are tabulated as in below. The table 2 given below shows the defect density value for processed input based on the number of defects that are generated and the lines of codes.

Table 2: Defect density

Number of defects	Lines of Code	Defect Density
9875	627727	0.016

The table 3 given below shows the fitness value obtained from our proposed modified ABC for various iterations. For different iterations the corresponding fitness value is estimated and the corresponding graph is plotted.

The fig 4 shows the graphical representation for the number of iterations and fitness values.

Table 3: Fitness value for different iterations

Iterations	Fitness Values
10	65.08
20	48.74
30	49.66
40	61.07
50	34.84

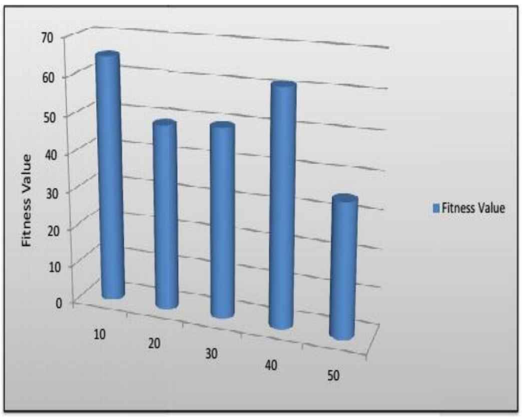

Figure 4: Graphical representation of fitness value for different iteration.

The table 4 given below shows the tabulation of various measures like specificity, sensitivity, execution time and accuracy corresponding to the different iterations. These values are calculated with the aid of the expressions mentioned above. These values provide performance evaluation of the particular system.

Table 4: Various measure for different iterations

Iterations	Specificity	Sensitivity	Execution time	Accuracy
10	0.193	1.0	19003	99.59
20	0.192	0.962	23359	99.57
30	0.133	0.637	20190	99.39
40	0.125	0.637	20164	99.38

The fig 5 given below shows the specificity of the proposed method. For various iteration values the corresponding specificity value is plotted.

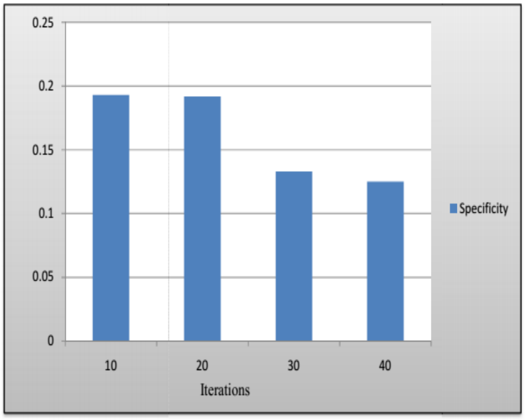

Figure 5: Graphical representation of specificity for different Iterations

The fig 6 given below shows the sensitivity value obtained in our proposed method. For different iterations the corresponding sensitivity value is plotted.

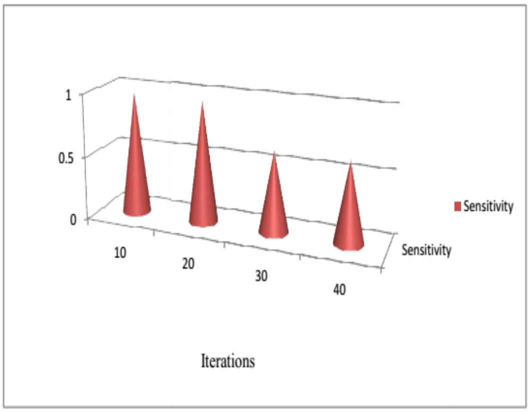

Figure 6: Graphical representation of sensitivity for different Iterations

The fig 7 given below shows the execution time of our proposed method. For different iterations the corresponding time is plotted.

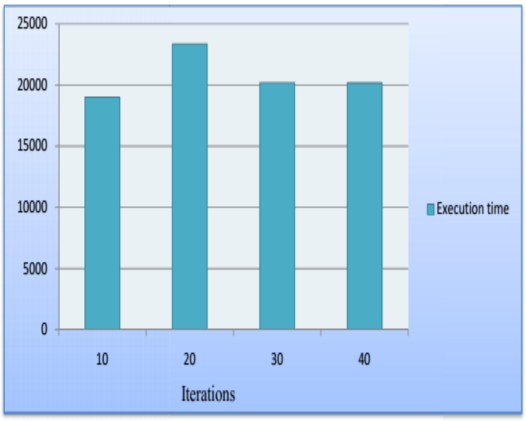

Figure 7: Graphical representation of Execution time for different Iterations

The fig 8 given below shows the accuracy obtained from our proposed method. For different iterations the corresponding accuracy is plotted.

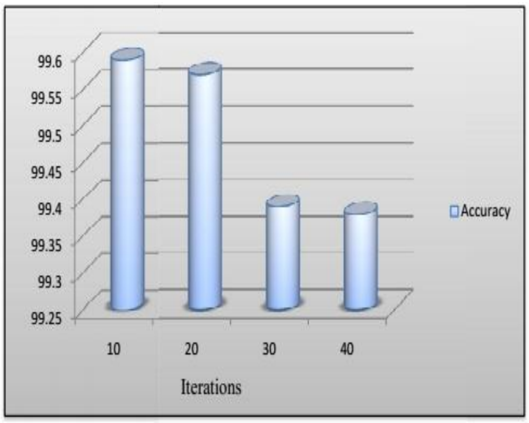

Figure 8: Graphical representation of accuracy for different Iterations

The fig.9 given below shows the accuracy value obtained from the proposed method and the existing method. The Exiting method referred here is the software defect classification based on Association rule mining, ABC algorithm and naïve bayes classifier. The value shows that our proposed method delivers better performance in terms of accuracy.

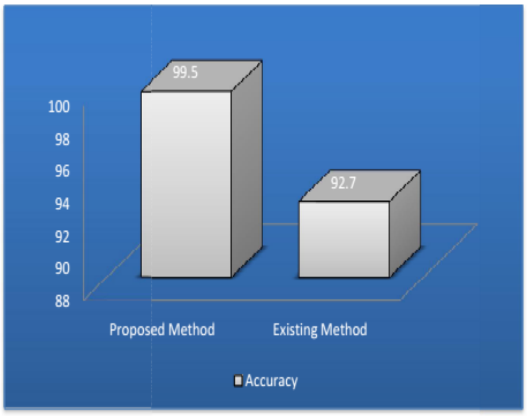

Figure 9: Graphical representation of Accuracy for proposed and existing method.

CONCLUSION

Defect classification using relational association rule mining based on fuzzy classifier along with artificial bee colony algorithm is proposed in this paper. At first features are selected from the database. Based on the features the rules are extracted using Relational Association Rule Mining (RARM). These rules are given as the input for Modified Artificial Bee Colony (MABC) algorithm to select the optimal rules. Then the best rules are fed to fuzzy classifier to predict the defect. The result of the proposed method shows that classification of defect in the rule was efficiently done when compared to the existing method. The proposed method achieves the 99.5% accuracy value which is high when compared to the existing method.

REFERENCE

[1] Mohamad Mahdi Askari and Vahid Khatibi Bardsiri, "Software Defect Prediction using a High Performance Neural Network",International Journal of Software Engineering and Its Applications,Vol. 8, No. 12,pp. 177-188,2014.

[2] Lin Lin, Mei-Ling Shyu and Shu-Ching Chen, "Association rule mining with a correlation-based interestingness measure for video semantic concept detection", Int. J. Information and Decision Sciences, 2009

[3] Pritam H. Patil,Suvarna Thube, Bhakti Ratnaparkhi and K.Rajeswari, "Analysis of Different Data Mining Tools using Classification, Clustering and Association Rule Mining" International Journal of Computer Applications, Vol.93, No.8, pp.0975–8887, 2014.

[4] Wanjiang. Han,Lixin. Jiang, Tianbo. Lu,Xiaoyan. Zhang and Sun Yi, "Study on Residual Defect Prediction using Multiple Technologies", Journal of Advances In Information Technology, vol. 5, no. 3, 2014.

[5] Hui Wang, "Software Defects Classification Prediction Based On Mining Software Repository",Thesis,2014

[6] Issam H. Laradji, Mohammad Alshayeb and Lahouari Ghouti, "Software defect prediction using ensemble learning on selected features",Information and Software Technology, 2014.

[7] Gabriela Czibula , Zsuzsanna Marian and Istvan Gergely Czibula, "Software defect prediction using relational association rule mining", Information Sciences, Vol. 264, pp.260–278,2014.

[8] Byoung-Jun Park , Sung-Kwun Oh and Witold Pedrycz, "The design of polynomial function-based neural network predictors for detection of software defects",Information Sciences, Vol.229,pp. 40–57, 2013.

[9] CH.Sekhar and S Reshma Anjum, "Cloud Data Mining based on Association Rule",International Journal of Computer Science and Information Technologies, Vol.5, No.2 ,pp.2091-2094,2014.

[10] Hafsa Zafar, Zeeshan Rana, Shafay Shamail and Mian M Awais, "Finding Focused Itemsets from Software Defect Data",In.proc.of 15th International Multitopic Conference (INMIC), 2012.

[11] Naheed Azeem and Shazia Usmani, "Analysis of Data Mining Based Software Defect Prediction

Techniques",Global Journal of Computer Science and Technology, Vol.11, No.16, 2011.

[12] S.Vijayarani and M.Sathiya Prabha, "Association Rule Hiding using Artificial Bee Colony Algorithm",International Journal of Computer Applications, Vol.33,No.2, 2011.

[13] Pooja Paramshetti and D. A. Phalke, "Survey on Software Defect Prediction Using Machine Learning Techniques",International Journal of Science and Research (IJSR),Vol.3, No.12, 2014.

[14] Remya Kartha K and Vikraman Nair R, "Data Mining for Causal Analysis of Software Defects",International Journal of Computer Science and Mobile Computing,pp.1-7,2013.

[15] Zsuzsanna Marian, "On The Software Metrics Influence In Relational Association Rule-Based Software Defect Prediction", Studia Univ. Babes Bolyai, Informatica, Vol.Lviii, No. 4, 2013

[16] T. Karthikeyan and N. Ravikumar, "A Survey on Association Rule Mining",International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3,No. 1, 2014

[17] Safia Yasmeen, "Software Bug Detection Algorithm using Data mining Techniques", International Journal of Innovative Research in Advanced Engineering (IJIRAE) Vol.1,No. 5,2014.

AUTHOR BIOGRAPHY

B.Dhanalaxmi obtained her Bachelor’s degree in Computer Science and information technology from VREC, (JNTU) Then she obtained her M.Tech in Software Engineering from GNEC (JNTU) and pursuing her PhD in Computer Science Engineering with specialization in Software Engineering from JNT University . Currently, she is an Associate Professor at Institute of Aeronautical Engineering, Hyderabad in the department of information technology. Her areas of research include software engineering, software testing, artificial neural network and databases.

Dr.G.Apparao Naidu, obtained his M. tech (Computer Science and Technology) from Andhra University in 2001.Then he obtained Phd (Computer Science and Engineering) from JNTU Kakinada in 2011 ,currently he is Professor In Department of Computer Science and Engeneering, Jbiet, Moinabad, Hyderabad. His areas of interests includes network security, Data mining, cloud computing, Software engineering, Software Testing.

Dr.K.Anuradha, obtained her M.tech (Computer Science and Technology) from Birla Institute Of Technology ,Ranchi in 2001.Then she obtained PhD (Computer Science and Engineering) from JNTU Anantapur in 2011 and obtained PhD (Mathematics) in 2006 ,currently she is Professor and Vice-Principal In Department of Computer Science and Engineering, GRIET, Hyderabad. Her areas of interests includes, Data mining, Software engineering, Compiler Design, Mathematical Foundations Of Computer Science.