International Journal of Engineering & Technology, 7 (1.1) (2018) 456-460

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

An optimized feature selection using fuzzy mutual information based ant colony optimization for software defect prediction

G. Manivasagam [1] *, R. Gunasundari [2]

[1] *Ph. D Research Scholar, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, Tamil nadu, India*
[2] *Associate Professor, Department of Information Technology, Karpagam Academy of Higher Education, Coimbatore, Tamil nadu, India*

Abstract

In recent years, there is a significant notification focused towards the prediction of software defect in the field of software engineering. The prediction of software defects assist in reducing the cost of testing effort, improving the process of software testing and to concentrate only on the fault-prone software modules. Recently, software defect prediction is an important research topic in the software engineering field. One of the important factors which effect the software defect detection is the presence of noisy features in the dataset. The objective of this proposed work is to contribute an optimization technique for the selection of potential features to improve the prediction capability of software defects more accurately. The Fuzzy Mutual Information Ant Colony Optimization is used for searching the optimal feature set with the ability of Meta heuristic search. This proposed feature selection efficiency is evaluated using the datasets from NASA metric data repository. Simulation results have indicated that the proposed method makes an impressive enhancement in the prediction of routine for three different classifiers used in this work.

Keywords: *Software Defect Prediction; Fuzzy Mutual Information; Ant Colony Optimization; Potential Features.*

1. Introduction

In the information era all the applications are software oriented and there is a rapid growth of the software development in all type of platforms. This arises the need to maintain the software quality more precisely. Most of the software's fail in their performance because of the poor designing and maintenance compatibilities. The software error or fault occurs mainly due to the human operation while developing coding or due to the computer system programs or incompatible OS versions.

Software testing process is introduced for early identification of software faults since the corrections in maintenance phase costs increase exponentially if defects are detected in later stages of SDLC. Moreover, it is important to note that the software testing alone covers sixty percent of the total software development expenditure. Therefore, testing the right modules is a crucial aspect.

The significant effect of identification and locating the defects in the module in software projects is a very complicated task. The larger the software project, the task of identification gets more Complex with easiness in testing and mechanism of evaluation does tremendously increase the cost of the task that software evaluation continuously increases and in a regimented manner and accurate estimation of project tasks which has a significant improvisation in the product and process qualities

The main motive of this proposed method is to predict the software defect and this is done by the evolutionary approach. The feature reduction is also a major fact to optimize the performance of the proposed method. This paper concentrates on two different issues namely feature selection and prediction of the software defects. The dataset used in this work is collected from the NASA dataset which contains the software defect details of the satellites. The voluminous dataset is handled by proposing a very effective feature selection method which produces only the relevant attributes for further processing instead of considering the whole dataset. The feature set which are selected and further used for prediction technique using ant colony optimization technique which is developed based on the inspiration of the ant's nature of predicting their food sources. The experimental results produce optimal result on software defect prediction

2. Related works

In [1] the authors mainly concentrated on the performance comparison of different methods available and an understanding of where appropriately this algorithm has to be applied using the NASA MDP datasets. In the work [2]the authors discussed about the various data mining approaches namely association rule mining, clustering and classification of the field of software defect prediction. This analysis process assists the software developer to predict the kind of software defect in a timely manner. Phalk, and Pooja [3] in their work adapted the concept of association rule discovery process to find the frequent pattern for software entities detection like the defects present in the software system.

Shukla and Deepak [4] examined different literates based on software defect prediction and in their work they drew a summarization of these existing approaches. The aim of the work in [5] was the development of models for predicting the software errors in order to measure the earlier stages of the software development life cycle to generate reasonable cost determination of a newly developed software system with best quality. The main theme of the paper [6] was to assist the developers to improvise the quality

of the software by predicting the software defects using the software metrics and using classification techniques.
In the work [7] the authors proposed principal component analysis based feature selection and prediction of defects using the neural networks with the reduced feature set. They use AR1 dataset available in PROMISE repository and they showed in their result that the neural networks with and without feature selection performance. Mohammad [8] in his work developed a software defect prediction model using the ensemble based machine learning for feature subset selection [8]. A methodical appraisal on prediction of software defect using the data mining approaches was developed by Romi [9].
The work in [10] investigated the pitfalls of Apriori algorithm and it did the improvement on Apriori algorithm by minimizing the rules generated based on the different parameters. They used new approach to find 'n' best association rule on heuristic analysis.
KamaljitKaur (2012) [11] Prediction of fault proneness software models in the fall tips for the models were unavailable and it was a very big challenge in the field of software industry. T attempted to predict the false promise of your model and the labels or not present. They used in genetic algorithm for software defect prediction approach.
Mrs. AgastaAdline, Ramachandran. M (2014) [12] Performed data analysis of various software for defect prediction techniques and described some of the algorithms and its usages in their work. They found that the objective of the fault in prediction using data mining is to improve that software quality development process. By using this kind of technique the cost effectiveness and the time complexity considerably reduced. By using this proposed method at the software manager effectively, the Hello Kitty resources in a limited manner and the overall error rates of all the existing techniques are compared and advantage of the proposed method was analysed in detail.
Karpagavadivu. K, et.al. (2012) [13] Proposed new kernel methods which protects the number of software defects in the module. This proposed approach was based on every computer matrix of kernel which was based on the similarity among the sofa model system. Novel Cornell method has been compared and shown that it achieved compatible results in the traditional linear and rbf Kernels. Furthermore, the project software defect prediction approaches also compared with the existing techniques of defect detection methods in the literature like linear regression and IBK. It was observed that the prior to the test case for maintenance phase and the software developers can use this proposed method to easily predict the most defective models in the system and focusing on the primarily rather than texting each and every module in the software system. This can decrease the testing effects and the project cost automatically.
AhmetOkutan and OlcayTanerYıldız, (2013) [14] in their work compared various machine learning techniques for software defect prediction of object oriented metrics using the artificial neural network techniques. This approach was found to be best suited for software prediction in case of object oriented metrics and it mainly used minimised calculations compared to the other artificial intelligence techniques. It has greater representation capability and it is capable of performing very complicated functions.
Yajnaseni Dash, Sanjay Kumar Dubey, (2012) [15] studied various data mining techniques for predicting the Sorcerer's with the help of association mining, classification and clustering techniques in their work. Best techniques are the helpline assisting the software engineers to develop a better model to detect the presence or absence of the defects in case of unknown labels with the help of unsupervised techniques.
Ms. Puneet Jai Kaur, Ms. Pallavi, (2013) [16] they used to the rank the performance Optimization approach for forecasting the software model development. Franking learning approach was used under the model was to develop the existing work sir and the later study was for improving the performance of the software model. They concentrated on two different aspects of an essay novel application of rank learning approaches applied on Real world data sets and another one was comprehensive evaluation and comparison of learning method with other existing approaches for predicting the order of software models.
Xiaoxing Yang, et.al. (2014) [17] this study showed the optimization of the process and the method was namely ranking approval on the existing methods by proving its accuracy in a higher way. Software defect prediction was not a new thing in software engineering domain. To come out with the right defect prediction model, various related studies and approaches have been conducted.

3. Methodology

In this proposed software defect detection task, the issue of the selection of potential features can be represented as follows:
Let the original software defect detection dataset be SF which consist of q features, determine optimal subset which can be denoted as O, which consist of 'r' features (r<q, O⊂SF), in such a way that the accuracy of classification is greatly maximized.
The fuzzy based artificial ants which involves in the feature selection of the software defect prediction consist of the following terms:

- r is the total number of attributes that establish the original Dataset, SF = $\{sf_1, \ldots, sf_r\}$.
- To search through the software defect prediction dataset's feature space, the number of artificial fuzzy ants for finding the optimal features are represented by fra ants
- For each feature sf_i and its associated pheromone trail intensity is represented as δi,
- For each ant k, a list of subset of features selected by the corresponding is denoted as $O_k = \{o_1, \ldots, o_m\}$.

This proposed work uses both the wrapper based evaluation and filter based evaluation for finding the performance of the selected subsets and the performance of the each individual features of the software defect prediction dataset. For evaluating the performance of the subsets, the linear regression model based classifier is used to determine the overall performance and the fuzzy mutual information based ant colony optimization is used for the individual importance of the feature in the selected subset.
During the first iteration process, individual fuzzy ant picks randomly a feature subset of r features. Solitarily the best subsets represented as b, b<fra, will be used for updating the trail of the pheromone and it also influence the feature subsets of the software defect prediction dataset of the next iteration. In subsequent iterations, every individual artificial ant will start with 'r-p' features which are picked randomly from the previously selected b-best subsets, the variable 'p' holds its value ranges between 1 and r-1. Likewise the features presented in the best 'b' feature subsets will have high chances of appearing in the subsets of the following iterations. Though, it will still be possible for every artificial ant to consider the remaining features also. For a given artificial ant k, whose features are best and selected based on the previously gained knowledge form the trails of the pheromone and the importance of the individual feature with respect to the 'O_k' subset consist of the specific features which are already chosen by that particular ant. The Efficient Feature Selection Measure (EFSM) is used for this purpose and defined as:

$$EFSM_i^{O_k} = \begin{cases} \dfrac{(\delta_i)^{\eta}(IF_i^{O_k})^{\kappa}}{\sum_{g \notin O_k} (\delta_i)^{\eta}(IF_i^{O_k})^{\kappa}} & \text{if } i \notin O_k \text{, Otherwise } 0 \end{cases}$$

Where $IF_i^{O_k}$ is the local individual importance of the features f_i given the subset o_k. The parameters η and κ control the effect of pheromone trail intensity and individual feature importance respectively.
$IF_i^{O_k}$ is defined as:

$$IF_i^{O_k} = H(sf_i) + H(sf_k) - H(sf_i, sf_k)$$

Where H (sf_i) and H (sf_k) are fuzzy entropy values for sf_i and sf_k respectively and H (sf_i,sf_k) is a fuzzy joint entropy

H (sf_i) = -

$$\frac{1}{n}\sum_{f\in sf}[\mu_{sfi}(f)\log\mu_{sf_i}(f)+(1-\mu_{sf_i}(f_i))\log(1-\mu_{sf_i}(f))\log(1-\mu_{sf_i}(f_i))]$$

H (sf_k) = -

$$\frac{1}{n}\sum_{f\in SF}[\mu_{sf_k}(f)\log\mu_{sf_k}(f)+(1-\mu_{sf_k}(f))\log(1-\mu_{sf_k}(f))\log(1-\mu_{sf_k}(f))]$$

$$H(sf_i \cup sf_k) = \begin{array}{l} -\frac{1}{n}\sum_{f\in SF}[\mu_{sf_i}(f)\vee\mu_{sfk}(f)]\log[\mu_{sf_i}(f)\vee\mu_{sfk}(f)] \\ +[1-\mu_{sf_i}(f)\vee\mu_{sfk}(f)]\log[1-\mu_{sf_i}(sf)\vee\mu_{sfk}(sf)] \end{array}$$

$$\mu_{c,k}=\left(\frac{\left\|s\overline{f}_c-sf_k\right\|\lambda}{v+\in}\right)^{\frac{-2}{m-1}}$$

The fuzzy membership value of k^{th} feature for c^{th} class is represented as $\mu_{c,k}$ which is formulated by khushaba et al [18]. In this 'm' is the fuzzification coefficient constant and $\in > 0$ is a fractional value for avoiding singularity and λ represents the standard deviation for computing distances. $s\overline{f}_c$ denotes the means of the data objects that belong to the class C and the radius of the data v is represented as $\left\|s\overline{f}_c-sf_k\right\|\lambda$

Proposed algorithm for fuzzy ant colony optimization based feature selection in software defect prediction

1) Process of initial setup:
   - Set δi = ctand Δδi= 0, (i= 1, …,n), where ctis a constant and Δδ is the amount of change of pheromone trial quantity for feature sfi.
   - Describe the maximum number of iterations.
   - Express b, where the b-best subsets will influence the subsets of the next iteration.
   - State p, where r– p is the number of features, every artificial ant will start with in the second and following iterations.
2) For the first iteration,
   - For j = 1 to fra,
   - Randomly assign a subset of r features to 0j.
   - Go to step 4.
3) Choose the remaining p features for every artificial ant:
   - For rr= r– p + 1 to r,
   - For j = 1 to fra,
   - Given subset Ok, Choose feature sfithat maximizes $EFSM_i^{O_k}$
   - Ok= Ok∪{sfi}.
   - Find and Substitute the replicated subsets, if any, with arbitrarily chosen feature subsets.
4) Evaluate the selected subset of each ant using a chosen classification algorithm:
   - For q= 1 to fra,
   - Evaluate the Mean Square Error (MSEq) of the classification results obtained by classifying the features of Oq.
   - Based on the obtained value of MSE sort the feature subsets. Reset the MSE value, and store the corresponding subset of features.
5) Using the feature subsets of the best b ant:
   - For q= 1 to b,

$$\Delta\delta_i=\begin{cases}\dfrac{\max\limits_{h=1:b}(MSE_h)-MSE_q}{\max\limits_{t=1:b}\left(\max\limits_{h=1:b}(MSE_h)-MSE_t\right)} & sf_i\in O_q\\ 0 & otherwise\end{cases}$$

$\delta_i = \tau\delta_i + \Delta\delta_i$

Whereτ is a constant such that (1 - τ) signifies the evaporation of pheromone trails.

- For q = 1 to fra,

Randomly produce r– p feature subset for ant q,to be used in the next iteration, and store it in Oj.

6) If the number of iterations is less than the maximum number of iterations, go to step 3.

Output: Optimal Feature subsets for software defect prediction

Logistic regression classifier

In statistics, logistic regression, or logit regression, or logit model is a regression model where the Dependent Variable (DV) is categorical. This article covers the case of a binary dependent variable-that is, where the output can take only two values, "0" and "1", which represents the outcomes such as pass/fail, win/lose, alive/dead or healthy/sick. Cases where the dependent variable has more than two outcome categories may be analysed in multinomial logistic regression, or, if the multiple categories are ordered, in ordinal logistic regression.[2] In the terminology of economics, logistic regression is an example of a qualitative response/discrete choice model.

4. Experimental result

The simulation of the proposed work is developed using MATLAB Code. The datasets used for this simulation result is obtained from the NASA public MDP (Modular toolkit for Data Processing) repository. This is a public repository for NASA datasets. NASA datasets are composed of several static code attributes. Each dataset describes the attributes of each project properties such as size, number of modules, and the number of defects.

In this work, four different datasets namely PC1, PC2, PC3 and PC4 are used which describes the flight software project used for an earth orbiting satellite which is written in C language. The detailed description of each of these datasets are discussed as follows

PC1 Dataset information

The PC1 dataset consist of 1109 instances with 22 attributes. The attributes are Loc, v(g), ev(g), iv(g), N, V, L, D, I, E, B, T, lOCode, lOComment, lOBlank, lOCode And Comment, uniq_Op, uniq_Opnd, total_Op, total_Opnd, branchCount, Defects (class attribute)

PC2, PC3 and PC4 Dataset information

The PC2 dataset consist of 5589 instances with 37 attributes. The attributes are Branch_Count, CallPairs, LocCodeand Comment, LocComments, Condition Count, Cyclomatic_Complexity, CyclomaticDensity, DecisionCount, DecisionDensity.

DesignComplexity.DesignDensity.EdgeCount, EssentialComplexity, EssentialDensity, LocExecutable, ParameterCount, HalsteadContent, HalsteadDifficulty, HalsteadEffort, HalsteadErrorEst, HalsteadLength, HalsteadLevel, HalsteadProgTime, HalsteadVolume, MaintenanceSeverity, ModifiedConditionCount, MultipleConditionCount, NodeCount, NormalizedCylomaticComplexity, NumOperands, NumOperators, NumUniqueOperands, NumUniqueOperators, NumberOfLines, PercentComments, LocTotal, Defects {False, True}.

The PC3 Dataset consist of 1563 instances with 38 attributes and PC 4 dataset consist of 1458 instances with 38 attributes. Both the datasets consist of same attributes as PC2 but with one additional attribute LOC_Blank.

Performance of feature selection process on four different software defect prediction dataset

In this work, four different NASA software defect prediction dataset namely pc1, pc2, pc3 and pc4 is used.

Table 1: Performance Comparison on PC1 Dataset Based on Feature Subset Generated

PC1 Dataset	Features Selected	No. of Features
FMIACO	2,3,14,15,17	5
ACO	3,5,9,14,15,17,21	7
Genetic Search	2,3,5,8,14,16,19,20,21,22	10

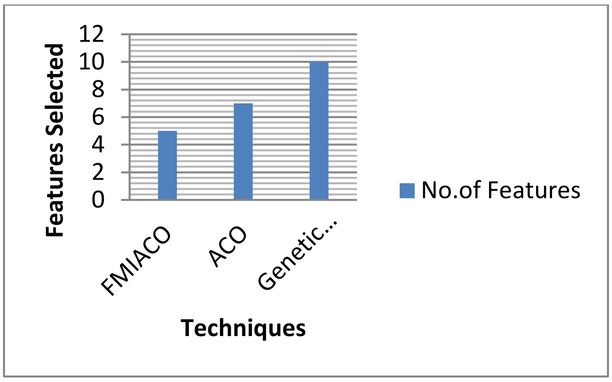


Fig. 1: Performance Comparison on PC1 Dataset Based on Feature Subset Generated.

The Table 1 and the Figure1 shows the performance of the proposed method fuzzy Mutual Information based artificial Ant Colony optimization based feature selection technique on PC1 NASA dataset with the existing standard Ant Colony optimization and Genetic Search. The output shows that the proposed FMIACO generates the least optimal feature of 5 features while the Genetic Search algorithm fails to choose best feature set due to its lack of local optimal search method.

Table 2: Performance Comparison on PC2 Dataset Based on Feature Subset Generated

PC2 Dataset	Features Selected	No. of Features
FMIACO	4,13,24,35	4
ACO	4,10,11,12,15,28	6
Genetic Search	2,4,5,7,9,11,13,15,17,19,22,36	12

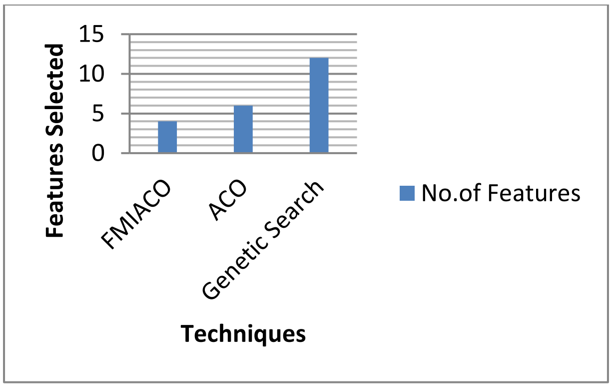


Fig. 2: Performance Comparison on PC2 Dataset Based on Feature Subset Generated.

The Table 2 and the Figure 2 shows the performance of the proposed method fuzzy Mutual Information based artificial Ant Colony Optimization based feature selection technique on PC2 NASA dataset with the existing Ant Colony Optimization and Genetic Search. The performance shows that the proposed FMIACO generates the less optimal feature subset of value 4 while ACO chooses 6 features as best features and Genetic Search chooses 12 attributes as the potential features from the whole feature set.

Table 3: Performance Comparison on PC3 Dataset Based on Feature Subset Generated

PC3 Dataset	Features Selected	No. of Features
FMIACO	1,4,5,18,25,26,33,36	8
ACO	1,5,7,9,23,26,25,29,31,33,36	11
Genetic Search	1,2,5,7,12,17,16,23,26,27,29,32,34,36	14

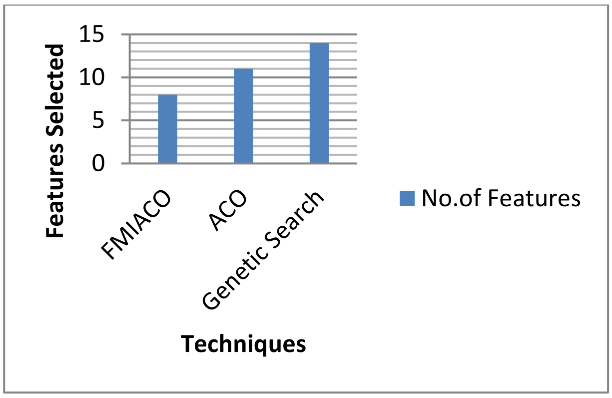


Fig. 3: Performance Comparison on PC3 Dataset Based on Feature Subset Generated.

The Table 3 and the Figure 3 shows the performance of the proposed method fuzzy Mutual Information based artificial Ant Colony Optimization based feature selection technique on PC3 NASA dataset with the existing Ant Colony Optimization and Genetic Search. The performance shows that the proposed FMIACO generates the less optimal feature subset of value 8 while ACO chooses 11 features as best features and Genetic Search chooses 14 attributes as the potential features from the whole feature set.

Table 4: Performance Comparison on PC4 Dataset Based on Feature Subset Generated

PC4 Dataset	Features Selected	No. of Features
FMIACO	4,6,17,36	4
ACO	1,3,7,9,24,29,31,33,36	11
Genetic Search	1,3,4,7,8,11,13,14,15,17,18,19,20,22,24, 25,26, 28,31,32,33,34,35,36,37	25

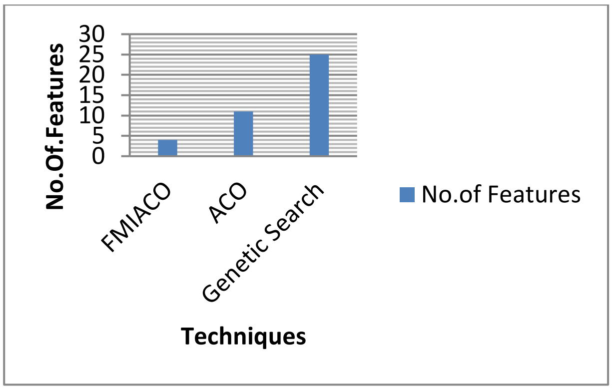
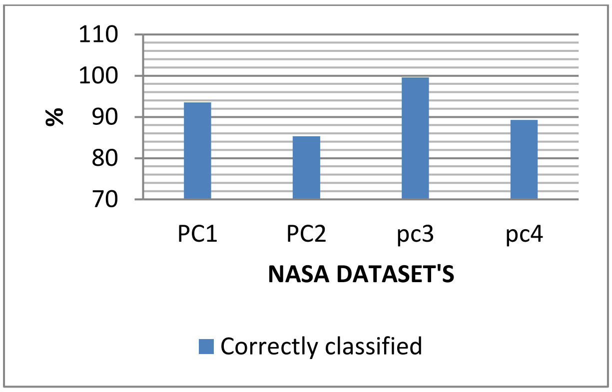


Fig. 4: Performance Comparison on PC3 Dataset Based on Feature Subset Generated.

The Table 4 and the Figure 4 shows the performance of the proposed method fuzzy Mutual Information based artificial Ant Colony Optimization based feature selection technique on PC4 NASA dataset with the existing Ant Colony optimization and Genetic Search. The performance shows that the proposed FMIACO generates the less optimal feature subset of value 4 while ACO chooses 11 features as best features and Genetic Search chooses 25 attributes as the potential features from the whole feature set.
The Table 5 shows the performance of the FMIACO based on the wrapper evaluation method of using linear regression classifier. As the capability to evaluate the individual importance of the features from the selected feature subset which greatly helps in improvising the performance of the proposed method for better classification and prediction of software defect in four different datasets.

Table 5: Performance of the Four Different Dataset with Linear Regression Classifier

Measures	PC1	PC2	PC3	PC4
Correctly classified	93.5077 (1037)	85.3021 (946)	99.5885 (5566)	89.3004 (1302)
Incorrectly classified	6.4923 (72)	4.6979 (163)	0.4115 (23)	10.6996 (156)
Mean absolute error	0.1161	0.2296	0.0082	0.1392
Root mean squared error	0.2431	0.3806	0.064	0.2792
Relative absolute error	89.33133	176.6863	97.67%	64.82%
Root relative squared error	95.617	149.737	99.9995 %	85.27%
TPR	0.993	0.912	1	0.936
FP rate	0.844	0.935	1	0.511
Precision	0.94	0.929	0.996	0.929
Recall	0.993	0.912	1	0.936
F measure	0.966	0.92	0.998	0.933



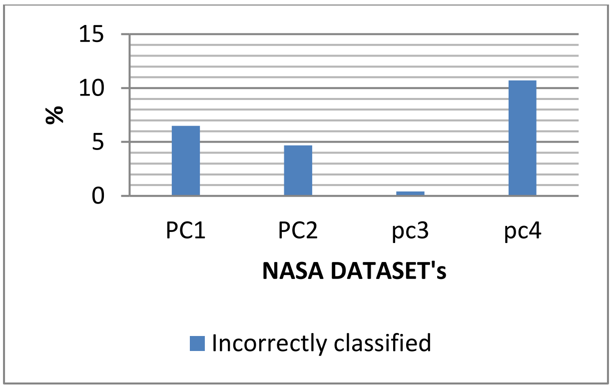


Fig. 5: Performance of the Four Different Dataset with Linear Regression Classifier.

The Figure 5 shows the correctly and incorrectly classified instances done by the linear regression model. The proposed work improvises the true positive rate of each datasets with the help of overall performance analysis of the feature subset and the individual importance of the features evaluated using the fuzzy mutual information in Ant Colony Optimization for software defect Prediction.

5. Conclusion

The two different major factors which affect the quality of the software defect prediction process are the imbalance of class and the irrelevant presence of attributes. An optimized method which integrates the filter and wrapper based approach of selection of potential feature subset form the four different NASA MDP datasets are introduced and their performance is compared with the other existing approaches and the overall performance of the selected optimal feature subset is validated using the linear regression classifier. The simulation results show that the proposed FMIACO achieves higher accuracy with the context of software defect prediction task. The fuzzy mutual information provides the degree of membership importance to each individual features with their corresponding class variables thus it exposes an impressive improvement in the prediction of software defects.

References

[1] Naresh E & Vijaya Kumar BP, "Comparative Analysis of the Various Data Mining Techniques for Defect Prediction using the NASA MDP Datasets for Better Quality of the Software Product", *Advances in Computational Sciences and Technology*, Vol.10, No.7, (2017), pp.2005-2017.
[2] PonPeriasamy AR & Mishbahulhud A, "Data Mining Techniques in Software Defect Prediction", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol.7, No.3, (2017).
[3] Paramshetti P & Phalk DA, "Software Defect Prediction for Quality Improvement Using Hybrid Approach", *International Journal of Application or Innovation in Engineering & Management*, Vol.4, No.6, (2015), pp.99-104.
[4] Shukla HS & Verma DK, "A Review on Software Defect Prediction", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, Vol.4, No.12), pp.4387-4394.
[5] Kumar Dwivedi V & Singh MK, "Software Defect Prediction Using Data Mining Classification Approach", *International Journal of Technical Research and Applications*, Vol.4, No.6, (2016), pp.31-35.
[6] Merugula S, "A Study on Software Defect Prediction Using Classification Techniques", *International Journal of Computer Science & Engineering Technology*, Vol.7, No.11, (2016).
[7] SatyaSrinivas M, Yesubabu A & Pradeepini G, "Feature Selection Based Neural Networks for Software Defect Prediction", *IOSR Journal of Computer Engineering*, Vol.18, No.6, (2016), pp.122-125.
[8] Laradji IH, Alshayeb M & Ghouti L, "software Defect Prediction using Ensemble Learning on Selected Features", *Information and Software Technology*, Vol.58, (2015), pp.388-402. https://doi.org/10.1016/j.infsof.2014.07.005.
[9] Wahono RS, "A Systematic Literature Review of Software Defect Prediction: Research Trends, Datasets, Methods and Frameworks", *Journal of Software Engineering*, (2015).
[10] Kaur A, "Defect Prediction by Pruning Redundancy in Association Rule Mining", *International Journal of Advanced Research in Computer Science*, (2017).
[11] Kaur K, "Analysis of resilient back-propogation for improving software process control", *International Journal of Information Technology and Knowledge Management*, Vol.5, No.2, (2012), pp.377-379.
[12] Adline A & Ramachandran M, "Predicting the Software Fault Using the Method of Genetic Algorithm", *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, Vol.3, Special Issue 2, (2014), pp.390-398.
[13] Karpagavadivu K, Maragatham T & Karthik S, "A Survey of Different Software Fault Prediction Using Data Mining Techniques Methods", *International Journal of Advanced Research in Computer Engineering & Technology*, Vol.1, No.8, (2012), pp.1-3.
[14] Okutan A & Yildiz OT, "A Novel Regression Method for Software Defect Prediction with Kernel Methods", *ICPRAM*, (2013), pp.216-221.
[15] Dash Y & Dubey SK, "Quality prediction in object oriented system by using ANN: a brief survey", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol.2, No.2, (2012).
[16] Kaur MPJ & Pallavi M, Data mining techniques for software defect prediction. *International Journal of Software and Web Sciences (IJSWS)*, Vol.3, No.1, (2013), pp.54-57.
[17] Yang X, Tang K & Yao X, "A learning-to-rank approach to software defect prediction", *IEEE Transactions on Reliability*, Vol.64, No.1, (2015), pp.234-246. https://doi.org/10.1109/TR.2014.2370891.
[18] Khushaba RN, Kodagoda S, Lal S & Dissanayake G, "Driver drowsiness classification using fuzzy wavelet packet based feature extraction algorithm", *IEEE Transactions on Biomedical Engineering*, Vol.58, No.1, (2011), pp.121-131. https://doi.org/10.1109/TBME.2010.2077291.
[19] http://promise.site.uottawa.ca/SERepository