Sampling-based Software Prone Technique for an Optimal Prediction of Software Faults

A. Balaram
[1]Ph.D. Scholar, Department of CSE, JNTUA University, Anantapur, Andhra Pradesh, India
balaram.balaram@gmail.com
S. Vasundra
[2] Professor, Department of CSE, JNTUA University, Anantapur, Andhra Pradesh, India
vasundras.cse@jntua.ac.in

Abstract

Finding faults in software modules is an emerging issue in software reliability systems and the assessment of the fault is performed by software fault prediction systems (SFPS). An identification process of fault-prone of software modules is the most prioritized before initiating the testing process of the same modules. The SFPS helps to improve the quality of software within the specified time and cost values. Early fault prediction in SFPS for the different software components showed significant results concerning the cost and time parameters. According to the state-of-the-art of SFPS, ensemble-based classifiers were performed as the best and most cost-effective when compared to other classifier methods. An ensemble random forest with adaptive synthetic sampling (E-RF-ADASYN) is developed recently is tested on a sample of PROMISE datasets and shows the cost-effective classifier results. Our proposed work is focused on the development of another sampling method, say, Multi-Distinguished-Features Sampling (MDFS) for obtaining the best sample illustration for representing the entire dataset. The experiments are conducted on bench-marked PROMISE datasets for demonstrating the efficiency of the proposed MDFS-E-RF compared to other traditional methods.

Keywords: **Software Reliability; Software Faults; Software Fault Predictions Systems; Ensemble Classifiers Sampling;**

1. Introduction

Assessment of software quality is the most important estimation of working reliability factor for the software products. Fault prediction in various components of software products is the primary scenario of the software quality assessment. It can be determined in early software development stages to reduce time and space requirements values. The fault prediction in software is commonly known as software fault prediction (SFP). The SFP is still facing some obstacles like fault density prediction due to software entity identification problems. Determining appropriate naming conventions is required and it is possible with the best representation of samples of data entities. Advances in software and its technologies are wildly used in fabulous social and real-life applications, such as air-traffic systems, space control systems, anonymous identification systems, defense systems, etc. Large-scale systems usually consist of many components, in such cases, - the various suspects or faults are a critical and challenging issue since they depend on the massive amount of data. Data characteristics or classifications for the components need to be assessed for the prediction of software faults. The ultimate aim of the prediction systems is to deliver an error-free and software-prone system. Classifier preserving techniques are developed and the best sampling strategy is designed in the proposed prediction system. Failure of the software is predicted at earlier stages using the best sample representations of components data. Thus, it became an error-free and optimal software-prone system. Developers have initiated the product models according to the client's data requirements. Clients' data are flushed as lakhs of records in daily life for large-scale applications. However, the classification of data is related to software components. Thus, multiple samples are extracted based on distinguishing features of software components and its proposed sampling schema is commonly known as Multi-Distinguished-Features Sampling (MDFS). The faults are also classified based on the software components. Based on the type of error, it easily detects the fault of a particular software component. With this proposed work, optimal predictions of software faults are derived which enable the efficient error-free software prone system. Imbalanced data learning is also one of the challenging issues in data mining applications. This issue manifests with two key steps: least preference interests and rare occurred instances. Machine learning algorithms are employed these two steps for accurate classification and minimal error rate. For example, financial and marketing applications need to recognize the fraudulent activities which are usually underlying with least preference people interests and also as rare occurred instances. These activities are classified and detected for effective handling of security for the

financial software systems. In biomedical data applications, occurring types of cancers are considered rare instances for people when compared to non-cancer cases. It is required to detect or classify the types of cancer in rare instances for the people underlying with the least preference interests. Because the cancers are found in people as unknown cases.

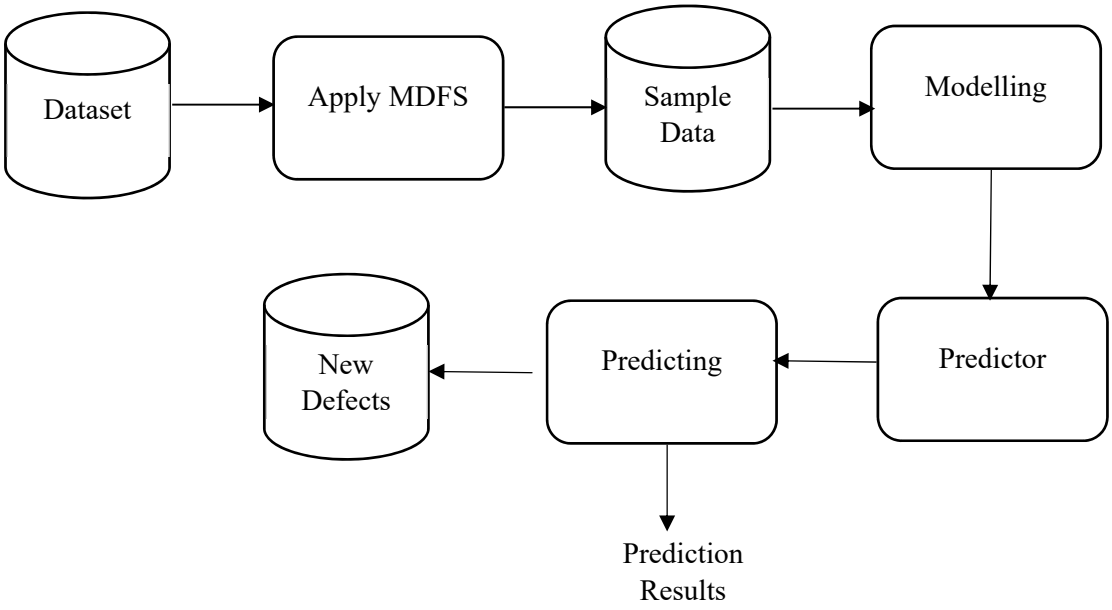

Fig. 1. Architecture of the Proposed Work

*I*ndustries are working on a large number of component data. Each software component log data is generated with thousands of entries. The inter-connected software systems generate a massive amount of log data. New defects or other software faults are assessed based on the large log data. Most of the log data was found with similar entries concerning software components. It is required to sample the massive amount of log data for improving the effectiveness of software systems to deliver the optimal software product systems. The log data is sampled based on finding the multi-distinguished features using our proposed schema MDFS. New defects and existing software faults are used for training the data while implementing the prediction system. Proposed MDFS covers the most and it is more effective than the ADASYN schema in finding the most distinguished samples. Thus, the MDFS-based ensemble random forest (MDFS-E-RF) produces the optimal predictions than E-RF-ADASYN. To obtain multi-distinguished features for the best sample illustration to assess the reliability of software with an optimized cost using proposed E-RF-MDFS.

Adaptative synthetic sampling delivers the samples of data randomized. It is unable to find the sample based on distinguishing features. Therefore, it sometimes fails to present the best sample illustration for large datasets. The proposed sampling technique, say, MDFS, initially, finds the distinguished software data objects, and then finds the samples near to derived distinguished data objects. Thus, it produces the best samples compared to the adaptive synthetic sampling method.

Contributions to the paper are summarized as follows

1. Develop the sample technique MDFS for deriving the best sample illustration
2. Design the hybrid classifier for making the optimal predictions
3. Enable the new defects in the training of predictions by implementing the incremental prediction system
4. Perform the modeling of log data for implementing the effective prediction system.
5. Conduct the performance study for demonstrating the effectiveness of the proposed work

The following sections describe the background work, proposed Multi-Distinguished-Features Sampling-based Ensemble Random Forest, experimental study and discussion, conclusion, and scope of the work respectively.

2. Background Work

Imbalanced data learning [1] is one of the crucial problems in software components or software products. Financial applications involve many entities' data; those data are regular or may also be fraudulent activity data. In such cases, predictions are required to find the suspected cases of fraudulent activities. If any software fault has

occurred, then there is less probability to detect the suspected cases. Fraudulent or suspicious logs are commonly happening in any industry or government sector applications. It is required to design the software system and able to classify the software faults under the classes of those activities. Many researchers designed various software fault prediction systems (SFPS). Recently, both the practical and other theoretical analyses are progressed on the real tracking software fault issue and much more impressed a continuous improvement implementation from both the fields of research and industry. It is shown in the establishment of various esteemed workshops and research platforms, including the International Conference on Machine Learning workshop on Learning from Imbalanced Data Sets (ICML'03) [2], AAAI [3], and the Association for Computing Machinery, which was focused on Knowledge Discovery and Data Mining explorations.

For handling the learning problems of imbalanced data, state-of-the-art strategies [4], [5], [6] are studied for the effective delivery of SFPS. Important methodologies are presented here which are useful for understanding the imbalanced learning problems and which are devised in three directions. (1) Sampling strategies. These strategies are aimed to design the two types of techniques, which are described for handling the oversampling and undersampling problems. These techniques are widely used for compensating the imbalanced data distributions for the obtained software log data. In this, [7], various curve-specific techniques were used to estimate the accuracy for the intermediate sample data i.e., it should not become underlying the cases of oversampling and undersampling. These are estimated with the decision tree learning classifier approaches. Advances in sampling are integrated with the decision probabilistic measures, scalable pruning, and initial data pre-processing [8]. Authors in [9] proposed another classifier schema "JOUS-Boost" for an effective solution to handle the learning of imbalanced data; which uses the combined learning techniques with the integration of adaptive boosting with jittering sampling techniques [10], [11].

(2) Synthetic data generation. It is the technique for overcoming the imbalanced data problem in a better way when compared to the earlier discussed method in this section. It generates the data samples artificially for the original imbalanced data. Authors in [12] proposed the SMOTE algorithm, which creates arbitrary examples that are related to examples of synthetic minority sample datasets. This includes the additional amount of shift towards the classifier learning bias technique toward the class of minority data objects. The SMOTE is some kind of Boost ensemble classifier technique and the extended work that is based on the technique is proposed in [13]. It is the procedure of synthetic procedure that can be combined with adaptive boosting techniques for transforming and updating the weights to learn the best compensation for the data cases of skewed data distributions. For ensuring the best accuracy in the case of classification for minority and majority data classes, another classification algorithm Data Boost-IM algorithm is proposed by the respective author of [14], in which the examples of synthetic data are created for both minority and majority classes [15], [16] through the use of "seed" samples.

(3) Cost-sensitive learning. The creative sampling strategies are used for the generation of synthetic data to simulate the learning process of imbalanced datasets cost-effectively. These techniques define a matrix which is called the cost matrix. The cost matrix handles distinct errors or instances for facilitating the effective learning of imbalanced data sets. It means that cost-sensitive learning techniques cannot change the intrinsic features of imbalanced distributions. The key target of this area of work is to create the required distinct cost matrices. These matrices are described to determine the misclassification of faults based on data characteristics of imbalanced data. One of the theoretical analyses to find optimal cost-sensitive learning for the classification of binary data which described [17]. In [18], the emerging idea of the instance weighting technique is used for reducing costs and improving the classification rate for imbalanced data classifications.

Data imbalance problems are solved with a combination of semi-supervised techniques and sampling strategies. Machine learning algorithms are much progressed in the last decade for designing SFPS. Turabiehet al.[19] have proposed the technique, Iterated feature selection approach combined with Layered Recurrent Neural Network (L-RNN) for deriving the classifier solutions of SFPS. Another requirement of this model is to define the specific computer model for enhancing the performance of software fault prediction using the relevant metrics. Other authors Tumaret al.[20]are designed and implemented an enhanced Binary Moth Flame Optimization (BMFO) with the sampling technique ADASYN. The technique of BMFO is explored with the initial step of wrapper feature selection and then sampling technique - ADASYN takes those features for the input dataset and the same is derived for obtaining the solution classification problems of imbalanced datasets.

Recently, Yucalar et al. [21]were derived a hybrid technique that combines the ensemble classifier technique and the Base Predictors, which framed as Rudin–Osher–Fatemi(ROF) with Multi-Layer Perception (MLP). This hybrid algorithm obtains improved performance results in the case of software predictions that have shown that reduced effort is enough for finding the faults in software components. The key limitation of this method is unable to perform software fault prediction for the irregular fault behaviour of software systems. In another recent method, authors, Qasem et al. [22]implemented CNN based Multi-Layer Perception (MLPs) system to overcome the problem of irregular fault behaviour of software component systems. Existing state-of-the-art of methods

effectively solve the issue of software faults classification problem; however, still, they created classifier optimization problems due to the reason of over fitting problems. It imposed imbalanced data to lower the accuracy of software fault classification and was unable to explore the greater number of software faults. The present techniques are designed in such a way using the MFDS for improving the classifier accuracy when compared to state-of-the-art existing methods.

3. Proposed Multi-Distinguished-Features Sampling-Based Ensemble Random Forest

For the best illustration of samples, the proposed schema of MDFS is presented in this section. The procedural steps are neatly indicated in Fig. 2.

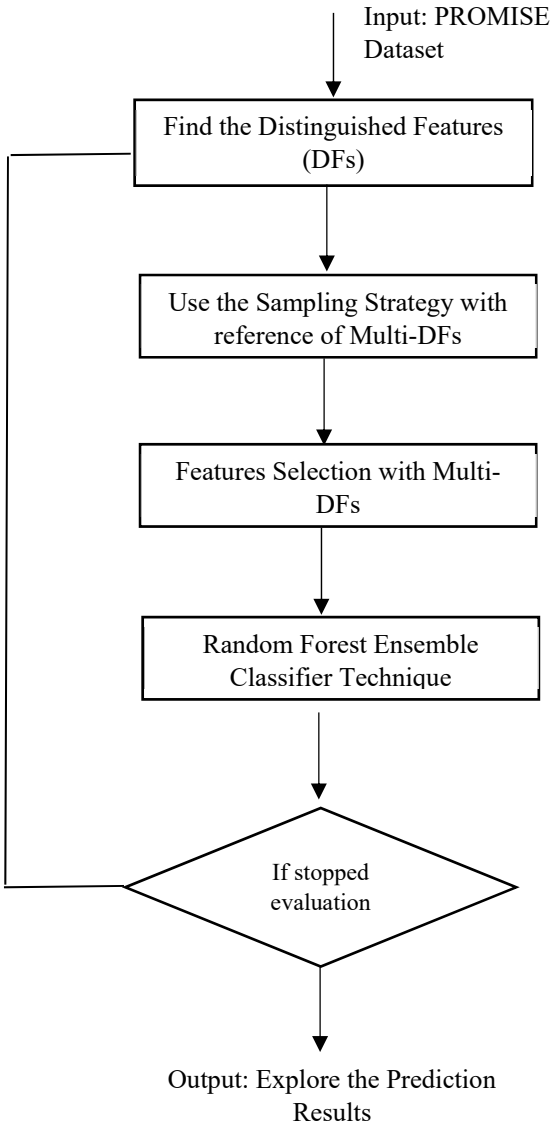

Fig. 2. Processing Steps of Proposed MDFS-E-RF.

The proposed work is developed with three key procedures, which are as follows: Find the multi-DFs using the sampling strategy, apply ensemble random forest until getting the satisfactory sample classifications, and derive the final prediction results.

Our proposed work intends to develop the ensemble classifier technique based on the best illustration of samples of original data to obtain optimal predictions. Two strategies are integrated to achieve for classifications of accurate software fault predictions, which are MFDS and ensemble random forest. This proposed algorithm is presented as follows:

Algorithm: Proposed MDFS-E-RF Technique

Input: PROMISE dataset

Output: Prediction Results

Methodology

Step 1: Find the DFs of the PROMISE dataset

Determine the initial most distinguished data object when compared to others

Find the number randomly 'r' over the n data objects of PROMISE data; Calculate the distances between among r^{th} data object to remaining other data objects and choose the index of most distinguished data to object based on finding the largest distance between the r^{th} data object and others as per Eqn. (1)

Step 2:Initialization of the data objects

Step 3: Determine other distinguished data objects based on the initial most distinguished data objects

Calculate the updated distances among the explored data objects and other remaining data objects
For i = 1 to N
$Distance_i$=minimum (largest_distance, distance ($PROMISEDATA_{max\_index}$, $PROMISEDATA_i$))

Step 4: Compute and Save other Most Distinguished Data Objects
Save the index of DFs is derived from argmax $_{I \epsilon \{1,2,..N\}}$ {$Distance_I$}, re-compute themaximum_index and maximum_dist
Steps 3 and 4 repeated to obtain and save multi DFs

Step 5: Apply the Ensemble Random Forest Classifier technique on Multi-DFs for optimal prediction results.

The classifier for the SFPS is derived and tested with the representations of samples. The best sample is derived with multi-distinguished features. Step 1 explains the idea of initial random object selection 'r' and determines the distances between the r^{th} data object and other data objects. One of the objects maintained the largest distance with r^{th} data object; in that case, the largest distance-maintained object is considered the most distinguished data object. Step 2 Describes the initialization of data objects. Step 3 describes distance updates with explored and non-explored data objects using the aggregate function of a minimum function. After updating the distances between explored and non-explored data objects, again the largest distance is derived using the aggregate function of maximum to save the index of another most distinguished data object. The function is shown in Eqn. (1)

$$\begin{cases} \text{largest}_{\text{index}} = \text{argmax I } \epsilon \{1,2,..\text{N}\} \\ \text{dist(PROMISEDATAr,} = \text{max\_dist} = \text{distance(PROMISEDATAr, PROMISEDATAI))} \end{cases} \quad (1)$$

A set of most distinguished data objects are derived finally and the nearest data objects are placed concerning multi-distinguished data objects. Apply the sample ratio and obtains an equal number of samples from every group. Step 4 is implemented to compute and save the most distinguished data objects and the maximum index is obtained. Step 5 determines the optimal predictions by applying of ensemble random forest classifier technique on Multi-DFs.

Ensemble Random Forest Classifier

The ensemble classifier takes the multiple classifiers under the unified trained model for deriving effective predictions. It has a good capacity for the classification or making predictions of software faults for the various components of software products. With the knowledge of multiple classifiers, it goes to assign a label to an unknown or unseen software tuple in the prediction process mechanism. It has used the ensemble classifiers KNN, Decision Tree (DT), and RF models. The sample strategy-based model i.e., E-RF-ADASYN is a very recent technique, and our proposed MDFS-based E-RF is developed to overcome the problem of best sample illustration. The best sample illustration is also played a vital role in ensemble random forest classifier results. For finding the best optimum classifier results of software faults for the products, the proposed work initially derives the best multi-DFs and then uses this data for ensemble random forest classification. The random forest follows the approach of ensemble and it can be improving the prediction accuracy when compared to E-RF-ADASYN. MDFS is the most superior to ADASYN in the case of finding the best sample illustrations of the original data. A decision tree is derived using the different samples or bootstraps which represent the original data. Using this bootstrap

sample, an optimal decision tree is constructed for finding effective decision makings for the software datasets. The decision of tree is carried out for each class object and it makes the vote for the selected forest has received the set of votes for various class objects. In this scenario, random forest (RF)uses the techniques of boosting and bagging for improving the success ratio for the prediction cases of software faults. The random forest features are described as follows:

In the RF, the generalization error is set based on important factors of the true strength. These factors are indicator function, tree of the forest, and size of the forest. It used the approach of maximum voting and the respective formula shown in Eqn. (2). It classifies the software faults in RF models based on elements shown in the same formula.

$$Proximity(i,j) = \frac{\sum_{t=1}^{n} K(a_t(i)=a_t(j))}{n} \quad (2)$$

Here, K() defines the indicator function for matching the classifier event and it becomes 0 or 1 depending on unmatching and matching cases respectively. a_t refers to the tree of the forest, and $a_t(i)$denotes the predictions for the ith data object. If Proximity (i,j) =1, then it indicates the matching of classes ith and jth predictions. Thus, the random forest provides the best ranks to the features which are the software variables. Once the evaluation of the classifier is completed stop the process. If the evaluation of the classifier is not completed again continue with the feature selection process. The efficiency of the proposed work for the benchmarked datasets is illustrated in the following experimental section.

4. Experimental Study And Discussion

Software faults classification is performed with state-of-the-art of classifier algorithms and proposed techniques. Experimental comparative analysis of the work is demonstrated with the performance measures, sensitivity, specificity, and area under the curve (AUC). The formulas of these performance measures are shown in Eqn. (3) to Eqn. (4).

A. Sensitivity

It measures the ratio of positives that defines as the ratio of correctly identified faults with affected condition (affected) to that of properly identified faults.

$$Sensitivity = (true\_pos)/(true\_pos + false\_neg\ ) \quad (3)$$

B. Specificity

It is the ratio of negatives that are properly recognized to that of improperly recognized conditions.

$$Specificity = (true\_neg)/(true\_neg + false\_pos\ ) \quad (4)$$

C. Area Under Curve

The Area Under Curve (AUC) is obtained between two limits that perform a definite integral between two points.

$$X=f(y) \quad (5)$$

Here, y=a and y= are integral limits;

The proposed MDFS-E-RF experiments and comparisons of existing algorithms are conducted for finding the solution of SFPS using the following combinations of configurations: i7 core and 16GB high configured desktop system and windows 10 operating system. Entire experimental demonstrations are conducted under the NVIDIA GTX edition GPX environment.

Experimental Analysis of Proposed Classifier Technique

The Promise dataset is available [23] and the same is used in this experimental study, here, a total of 20 attributes are taken, which are described as follows.

1. loc : count for the lines_of_code
2. ccomp : cyclomatic_complexity"
3. ecomp : essential_complexity"
4. designcomp : design_complexity"
5. totaln: count of operators and operands
6. vol: volume
7. pgml : program length

8. df: difficulty factor
9. intell: intelligence
10. ef: effort
11. te:time_estimation
12. lOCode : line count
13. lOComment : lines of comments
14. lOBlank :total number of blank lines
15. uniq_Op :unique_operators
16. uniq_Opnd :unique_operands
17. totalop:total_operators
18. totalOpnd:total_operands
19. branchCount : numeric % of the flow graph
20. defects : {false,true}

The classifier predicts the class label information defects based on the other attributes of the PROMISE datasets. The confusion matrix is widely used in finding the information of true pos, true neg, false pos, and false neg. Performance measures of the classifier used these four values of the confusion matrix. Table 1 presents these values in the form of a confusion matrix.

Prediction Vs Actual Class	**Class Prediction**	
	Yes-Class	**No-Class**
Actual Class	true_pos	false_neg
	false_pos	true_neg

Table 1. Confusion Matrix Values

Our proposed work MDFS-E-RF has experimented with fifteen different software faults using the PROMISE software engineering database. It is well-structured data and consists of no missing data. Table 2 shows the information of used datasets of PROMISE in this experimental study. This dataset is selected over the project consists of 20 features and takes one binary classifier output for representing the software faults value.

S. No	**Name of the Datasets**
1	Camel -1.0
2	Ant-1.7
3	Camel-1.4
4	Camel-1.2
5	Jedit-3.2
6	Camel – 1.6
7	Jedit-4.1
8	Jedit– 4.0
9	Jedit-4.2
10	log4j-1.0
11	Jedit-4.3
12	log4j-1.2
13	log4j-1.1
14	Xalan -2.7
15	Xalan-2.4

Table 2. Dataset Description.

Two existing and proposed MDFS-E-RF are experiments for the fifteen datasets for the predictions of software faults towards the software component systems. The comparative results are shown in Table 3. From these values, it was observed that our proposed ensemble classifier achieved the best classifier predictions when compared to others in the case of determining the software faults. Effective SFPS is obtained through the developments of MDFS and ensemble random forest classifiers.

S. No	Name of the Datasets	Decision Tree (DT) Classifier	k-nearest neighbour classifier (KNN)	Proposed MDFS-E-RF
Sensitivity				
1	Camel -1.0	0.421	0.514	0.654
2	Ant-1.7	0.423	0.508	0.622
3	Camel-1.4	0.345	0.499	0.564
4	Camel-1.2	0.362	0.585	0.678
5	Jedit-3.2	0.432	0.567	0.712
6	Camel – 1.6	0.444	0.511	0.645
7	Jedit-4.1	0.432	0.498	0.586
8	Jedit– 4.0	0.345	0.467	0.675
9	Jedit-4.2	0.389	0.486	0.665
10	log4j-1.0	0.398	0.523	0.643
11	Jedit-4.3	0.377	0.478	0.689
12	log4j-1.2	0.388	0.511	0.599
13	log4j-1.1	0.433	0.546	0.621
14	Xalan -2.7	0.322	0.453	0.567
15	Xalan-2.4	0.345	0.456	0.623
Specificity				
1	Camel -1.0	0.343	0.423	0.552
2	Ant-1.7	0.322	0.432	0.567
3	Camel-1.4	0.341	0.478	0.567
4	Camel-1.2	0.311	0.422	0.532
5	Jedit-3.2	0.389	0.488	0.612
6	Camel – 1.6	0.444	0.522	0.576
7	Jedit-4.1	0.411	0.523	0.610
8	Jedit– 4.0	0.323	0.498	0.622
9	Jedit-4.2	0.386	0.488	0.666
10	log4j-1.0	0.322	0.453	0.578
11	Jedit-4.3	0.372	0.477	0.681
12	log4j-1.2	0.382	0.521	0.579
13	log4j-1.1	0.333	0.526	0.632
14	Xalan -2.7	0.422	0.522	0.577
15	Xalan-2.4	0.445	0.566	0.635
AUC				
1	Camel -1.0	0.521	0.654	0.812
2	Ant-1.7	0.543	0.622	0.822
3	Camel-1.4	0.455	0.566	0.677
4	Camel-1.2	0.566	0.623	0.689
5	Jedit-3.2	0.567	0.713	0.856
6	Camel – 1.6	0.522	0.622	0.876
7	Jedit-4.1	0.532	0.675	0.876
8	Jedit– 4.0	0.445	0.667	0.775
9	Jedit-4.2	0.589	0.686	0.765
10	log4j-1.0	0.521	0.689	0.845
11	Jedit-4.3	0.477	0.578	0.789
12	log4j-1.2	0.488	0.611	0.799
13	log4j-1.1	0.533	0.722	0.876
14	Xalan -2.7	0.433	0.567	0.721
15	Xalan-2.4	0.521	0.578	0.876

Table 3. Sensitivity, Specificity, and AUC Values for the Existing and Proposed Methods.

Overall comparisons of performance values, i.e., sensitivity, specificity, and AUC cases, proposed MDFS-E-RF achieved the best values compared to the other two existing techniques. It is tested for the distinct subsets of PROMISE datasets in analyzing the software faults classification. Existing DT and KNN are shown as not many impressive methods as per the empirical analysis provided in Fig. 3, Fig. 4, and Fig. 5.

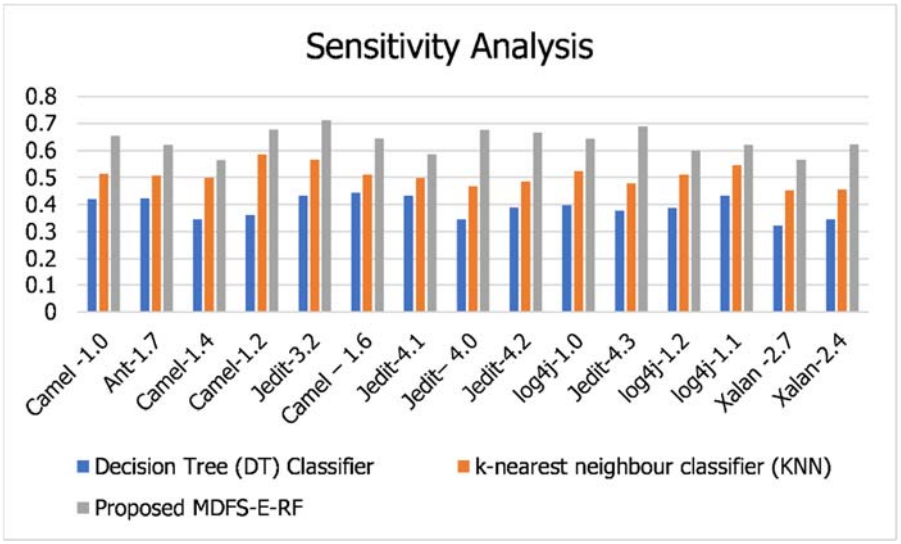

Fig. 3. Sensitivity Comparative Analysis.

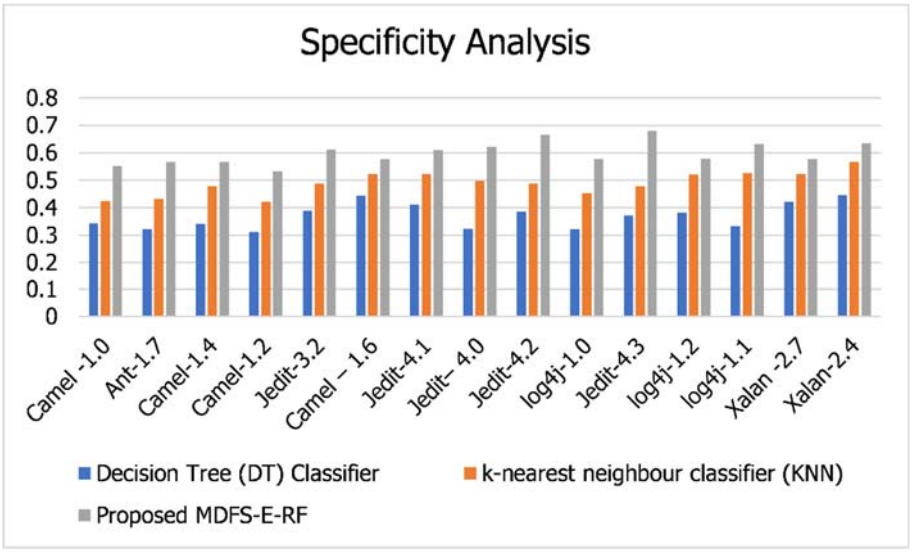

Fig. 4. Specificity Comparative Analysis.

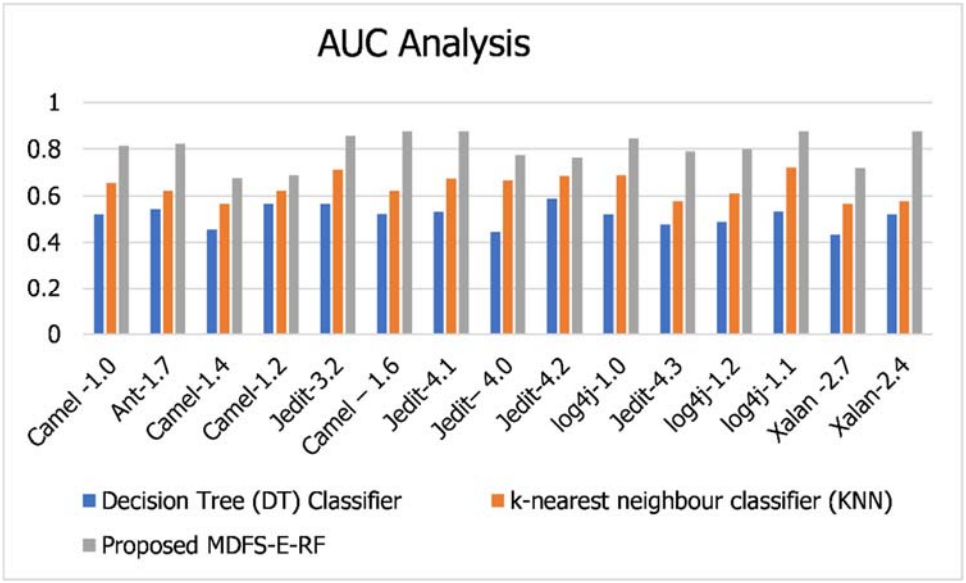

Fig. 5. AUC Comparative Analysis.

One of the performance measures of the classifier, that is, AUC, its calculation depends on the ratio of true pos and false pos rates. The confusion matrix is initially derived with the classifier values which have shown the four values as per Table 1. and Table 3 shows the Sensitivity, Specificity, and AUC performance measures values which are derived using feature extraction of given fifteen datasets. The performance of the proposed MDFS-E-RF achieved the best for the case all given fifteen software datasets while the classification of software faults. As per the experimental investigations, it was found that less accurate prediction results are derived with DT when compared to KNN. The KNN improves the classification results when compared to DT. However, it also shows required accurate classifier results with original datasets. Thus, our proposed work finds the best sample illustrations of original data using the MDFS and uses the ensemble random forest classifier. Therefore, the same observations are made when the investigation of experimental results of existing and proposed techniques.

5. Conclusion and Scope of the work

The proposed work in this paper is focused on the intelligent way of finding the best sample illustrations for making efficient predictions of software faults. Relevant sample features are sufficiently drawn using the technique of MDFS in the present work. It resolves the problem of best sample illustrations of data, unlike other sampling strategies. The PROMISE data is carried out in our experimental work in which, fifteen different subsets of datasets are collected based on data metrics of the software. These datasets are used in the experimental study for illustrating the classifier's accuracy in finding the software faults. These experimental investigations proved that our proposed MDFS-E-RF is outperformed compared to other techniques, like DT and KNN classifier techniques. The future scope of the work is to develop scalable classifier techniques using big data technologies for large datasets to overcome the problems of time and space complexities.

Conflicts of interest

"The authors have no conflicts of interest to declare"

References

[1] Vluymans, S. (2019). Learning from Imbalanced Data. In: Dealing with Imbalanced and Weakly Labelled Data in Machine Learning using Fuzzy and Rough Set Methods. Studies in Computational Intelligence, vol 807. Springer, Cham.
[2] Chawla, N.V.; Japkowicz, N.; Ko lcz, A.(Ed.), Imbalanced Clustering for Microarray Time-Series, in Proc. ICML'03 Workshop on Learning from Imbalanced Data Sets, 2003.
[3] Japkowicz, N. (Ed.). Learning from Imbalanced Data Sets, the AAAI Workshop, Technical Report WS-00-05, American Association for Artificial Intelligence, 2000.
[4] Rathore, S.S.; Kumar, S. (2021). Software fault prediction based on the dynamic selection of learning technique: findings from the eclipse project study. *Appl Intell* **51,** 8945–8960.
[5] Matloob, F. *et al*. (2021). "Software Defect Prediction Using Ensemble Learning: A Systematic Literature Review, in *IEEE Access*, vol. 9, pp. 98754-98771.
[6] Cetiner, M.; Sahingoz, O.K. A Comparative Analysis for Machine Learning based Software Defect Prediction Systems," *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020, pp. 1-7, Drummond. C. and Holte, R. C4.5, Class Imbalance, and Cost Sensitivity: Why Under-sampling Beats Oversampling, in Proc. ICML'03 Workshop on Learning from Imbalanced Data Sets, 2003.
[7] Chawla, N. (2003). C4.5 and Imbalanced Datasets: Investigating the Effect of Sampling Method, Probalistic Estimate, and Decision Tree Structure," in ICML-KDD'03 Workshop: Learning from Imbalanced Data Sets.
[8] Mease, D.; Wyner, A.J.; Buja, A. (2007). Boosted Classification Trees and Class Probability/Quantile Estimation, Journal of Machine Learning Research, vol. 8, pp. 409- 439.
[9] Basha, M.S.; Mouleeswaran, S.K.; Prasad, K.R. (2021). Sampling-based visual assessment computing techniques for an efficient social data clustering. *J Supercomput* **77,** 8013–8037.
[10] Rajendra Prasad, K.; Mohammed, M.; Narasimha Prasad, L.V.; *et al.* (2021). An efficient sampling-based visualization technique for big data clustering with crisp partitions. *Distrib Parallel Databases* **39,** 813–832.
[11] Chawla, N.V.; Hall, L.O.; Bowyer, K.W.; Kegelmeyer, W.P. (2002). SMOTE: Synthetic Minority Oversampling Technique, Journal of Artificial Intelligence Research, vol. 16, pp. 321-357.
[12] Chawla, N.V.; Lazarevic, A.; Hall, L. O.; Bowyer, K.W. (2003). Smoteboost: Improving Prediction of the Minority Class in Boosting, in Proc. European Conf. Principles and Practice of Knowledge Discovery in Databases, pp. 107-119, Dubrovnik, Croatia.
[13] Guo, H.; Viktor, H.L. (2004). Learning from Imbalanced Data Sets with Boosting and Data Generation: the DataBoost-IM Approach, in SIGKDD Explorations: Special issue on Learning from Imbalanced Datasets, vol.6, issue 1, pp. 30 - 39.
[14] Balaram, A.; Vasundra, S. (2022). A Review on Machine Learning Techniques to Predict the Reliability in Software Products. In: Gunjan, V.K., Zurada, J.M. (eds) Proceedings of the 2nd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities, and Applications. Lecture Notes in Networks and Systems, vol 237. Springer, Singapore. https://doi.org/10.1007/978-981-16-6407-6_28
[15] Balaram, A.; Vasundra, S. (2022). Prediction of software fault-prone classes using ensemble random forest with adaptive synthetic sampling algorithm. *Autom Softw Eng* **29**.
[16] Elkan, C. (2001). The foundations of cost-sensitive learning, in Proc. Int. Joint Conf. Artificial Intelligence (IJCAI'01), pp. 973-978.
[17] Ting, K.M. (2002). An instance-weighting method to induce cost-sensitive trees, IEEE Transaction on Knowledge and Data Engineering, 14: pp. 659-665.
[18] Tumar; Hassouneh, Y.; Turabieh, H.; Thaher, T. (2020). Enhanced Binary Moth Flame Optimization as a Feature Selection Algorithm to Predict Software Fault Prediction, in *IEEE Access*, vol. 8, pp. 8041-8055.

[19] Tumar; Hassouneh, Y.; Turabieh, H.; Thaher, T. (2020). Enhanced Binary Moth Flame Optimization as a Feature Selection Algorithm to Predict Software Fault Prediction," in *IEEE Access*, vol. 8, pp. 8041-8055.
[20] Fatih Yucalar; Akin Ozcift; Emin Borandag; Deniz Kilinc (2020). Multiple-classifiers in software quality engineering: Combining predictors to improve software fault prediction ability, Engineering Science and Technology, an International Journal, Volume 23, Issue 4.
[21] Qasem, O.A.; Akour, M.; Alenezi,M. (2020). The Influence of Deep Learning Algorithms Factors in Software Fault Prediction, in *IEEE Access*, vol. 8, pp. 63945-63960.
[22] promise.site.uottawa.ca/SERepository/datasets/cm1.arff.
[23] Bal, P.R.; Kumar, S. (2020). "WR-ELM: Weighted Regularization Extreme Learning Machine for Imbalance Learning in Software Fault Prediction, in *IEEE Transactions on Reliability*, vol. 69, no. 4, pp. 1355-1375.

Authors Profile

Mr. A. Balaram working as an Associate Professor in the Dept. of Computer Science and Engineering, CMR Institute of Technology, Hyderabad. He obtained his M. Tech from JNTUA University and B. Tech from JNTUH University India. Pursuing Ph.D. in JNTUA University Anantapur. He has more than 15 years of teaching experience He has published in 28 international Journals, 11 conferences, one book chapter, and two text books. And having three patents. His research interests are Software Engineering, Network Security & Cryptography, Machine Learning, and Cloud Computing.

Dr.S.Vasundra, Professor of Department of Computer Science and Engineering and NSS Coordinator, JNT University, Anantapuramu.She obtained her M.Techand Ph.D.degree from JNTUA University and B.E degree from Gulbarga University. She has published in more than 59 international journals, 21 conferences, and one text book. And also has three Patents. Her research interests include Mobile Ad hoc Networks, Computer Networks, Big Data, data mining, and cloud computing.