International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I7P201
Volume 70 Issue 7, 1-9, July 2022
© 2022 Seventh Sense Research Group®

Original Article

Efficient Dimensionality Reduction using Improved Fuzzy C-Means Entropy Approach with Caps-TripleGAN for Predicting Software Defect in Imbalanced Dataset

Satya Srinivas Maddipati[1], Malladi Srinivas[2]

[1,2]*CSE Department, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India.*

[1]maddipativas@gmail.com

Received: 12 May 2022 Revised: 15 June 2022 Accepted: 25 June 2022 Published: 18 July 2022

Abstract *- Early detection of bugs or defects in software life cycle reduces the effort required to develop software. Mainly there are two problems in software defect prediction, i.e., dimensionality reduction and class imbalance. Several works are done to predict software defects, but that method does not provide sufficient accuracy, and the error rate increases. To overcome these issues, this work is proposed. In this manuscript, Improved Fuzzy C-means-based Entropy (IFCME) approach with CapsNet triple generative adversarial network (Caps-Triple GAN) for predicting software defect imbalanced dataset for reducing dimensionality and Class Imbalance Problem. Converting the non-linear high dimensional data into low dimensional space to reduce the class imbalance problem uses IFCME. Caps-Triple GANs are used to classify the data with high accuracy and reduce the error rate of software data prediction. The simulation process is executed in the MATLAB platform. The proposed IFCME-Caps-Triple GAN-DR-SDP attains higher accuracy of 23.84%, 32.94%, 36.94%, High Precision of 26.94%, 37.32%, 28.94%, and the proposed method is compared with the existing methods such as Software defect prediction model based on LASSO–SVM (LASSO–SVM-DR-SDP), multi-source heterogeneous cross-project defect prediction via multi-source transfer learning and autoencoder (MHCPDP-MAA-DR-SDP), Tackling class imbalance problem in software defect prediction through cluster-based over-sampling with filtering (KMFOS-DR-SDP)respectively.*

Keywords *- Improved Fuzzy C-means-based Entropy (IFCME), Software defect prediction, Caps Net triple generative adversarial network (Caps-Triple GAN), Dimensionality reduction, and Class Imbalance Problem.*

1. Introduction

Nowadays, the need for software is increasing due to technology's growth. While more people are using the software, some defects may occur in the system that will affect the software, and many errors enter the system [1-3]. Software testing plays a significant role in developing the software [4-6]. While developing the software, there is a chance of affecting unexpected errors. So the software get fails and needs some maintenance process [7]. Due to this reason, testing plays a vital role in the development of software. There are mainly two problems in SDP: dimensionality reduction and class imbalance [8]. Several works are done to predict software defects, but that method does not provide sufficient accuracy, and the error rate increases [9-10]. To overcome these issues, this work is proposed.

In this manuscript, Improved Fuzzy C-means-based Entropy (IFCME) [11] Approach with CapsNet triple generative adversarial network (Caps-Triple GAN) [12] for predicting software defect imbalanced dataset for reducing dimensionality reduction and Class Imbalance Problem. Converting the non-linear high dimensional data into low dimensional space to reduce the class imbalance problem uses IFCME. Caps-Triple GAN is used to classify the data with high accuracy and reduce the error rate of software data prediction.

The main contributions are summarized below,

- This manuscript Improved Fuzzy C-means-based Entropy (IFCME) Approach with Caps Net triple generative adversarial network (Caps-Triple GAN) for predicting software defect imbalanced dataset for reducing dimensionality and Class Imbalance Problem.
- Converting the non-linear high dimensional data into low dimensional space to reduce class imbalance problems uses IFCME.
- Caps-Triple GAN is used to classify the data with high accuracy and reduce the error rate of software data prediction.

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

- The simulation process is executed in the MATLAB platform.
- The software defect is predicted using the two datasets such as NASA [13] and AZEEEM [14].
- Then the performance of the proposed IFCME-Caps-Triple GAN-DR-SDP method is measured in terms of accuracy, recall, precision, F-measure, and error rate.
- Then, the efficiency of the proposed method is compared with existing methods such as LASSO–SVM-DR-SDP [21], MHCPDP-MAA-DR-SDP [22], KMFOS-DR-SDP [23], respectively.

The remaining manuscript is mentioned below. Section 2 delineates that literature survey. Section 3 Proposed methods for software prediction using IFCME- Caps-Triple GAN. Section 4 illustrates the result and discussion. At last, section five concluded the manuscript.

2. Literature Survey

In 2021, Wang et al. [21] presented the SDP model based on LASSO–SVM. At first, the feature selection ability of minimum absolute value compression and selection technique was used to lessen the dimension of the unique dataset. The dataset was not related to software defect prediction. Then, the optimal value of the support vector machine is obtained using the parameter optimization ability of the cross-validation algorithm. At last, software defect prediction was completed using the ability of non-linear computing of support vector machine. Then, the accuracy of the simulation results was 93.25%.

In 2021, Wu et al. [21] presented the multi-source heterogeneous cross-project defect prediction through the multi-source transfer learning and autoencoder. In this manuscript, an HCPDP technique called MHCPDP was used. Besides, incorporating multiple source projects increases the number of source datasets, and multi-source heterogeneous cross-project defect prediction develops a multi-source transfer learning algorithm that reduces the effect of negative transfers and upgrades the efficacy of the classifier. The accuracy of simulation results was 89.09%.

Gong, Wu et al. [21] have presented the Tackling class imbalance problem in software defect prediction through cluster-based oversampling with filtering. This manuscript uses cluster-based over-sampling with noise filtering (KMFOS) method to tackle the class imbalance problem in SDP. Cluster-based over-sampling with noise filtering divides the defective instances into K clusters; new defective instances were generated by interpolating every 2 clusters. The accuracy of simulation results was were78.09%. In Jyothi G, Bal K[28] proposed two-layer ensemble learning to predict software defects.

3. The proposed method for software prediction using IFCME- Caps-Triple GAN

This manuscript Improved Fuzzy C-means-based Entropy (IFCME) Approach with CapsNet triple generative adversarial network (Caps-Triple GAN) for predicting software defect imbalanced dataset for reducing dimensionality and Class Imbalance Problem. The block diagram of the proposed method for software prediction using IFCME- Caps-Triple GAN is shown in Figure 1. IFCME is used to convert the non-linear high-dimensional data into low-dimensional space and to reduce the class imbalance problem. Caps-Triple GAN is used to classify the data with high accuracy and reduce the error rate of software data prediction.

3.1. Data Acquisition

Two datasets are taken to detect the software detection problem from the imbalanced dataset. The datasets are 1. NASA Dataset, this dataset is represented as Soft lab. The soft lab is the Software Research Laboratory at Bogazici University, Istanbul, Turkey. Level static code attributes' functions are collected using pretest metrics extraction and analysis tools. In this data, there are two types of datasets; one will discretely keep defect information, whereas the other will keep the bug count associated with defectiveness. The second dataset is the AEEEM dataset, which contains 61 software metrics: 17 source code metrics, seventeen entropy of source code metrics, and seventeen churns of source code metrics. These datasets are imbalanced. An imbalanced dataset means most of the samples are non-defective, and extremely low samples are defective. So, the classifiers are biased because of the imbalanced nature of the data. Data imbalance is reduced with the help of the under-sampling or oversampling process. So these datasets are pre-processed to decrease the dimensionality reduction and data cleaning for converting imbalanced datasets into balanced datasets.

3.2. Improved fuzzy c-means-based entropy (IFCME) for pre-processing

In this, IFCME is used for Pre-processing. Here, the pre-processing stage is applied for data cleaning to decrease dataset imbalance and dimensionality reduction from the input dataset. The input dataset consists of noise removed in the data cleaning process.

Then the dataset dimensions are reduced using the multi-factor 2×2 possibility table consisting of $K(\eta K)/M(\eta M)$ based on fuzzy groups with controls and cases. To reduce dimensionality and the class imbalance problem in the dataset, the IFCME consists of 5 stages.

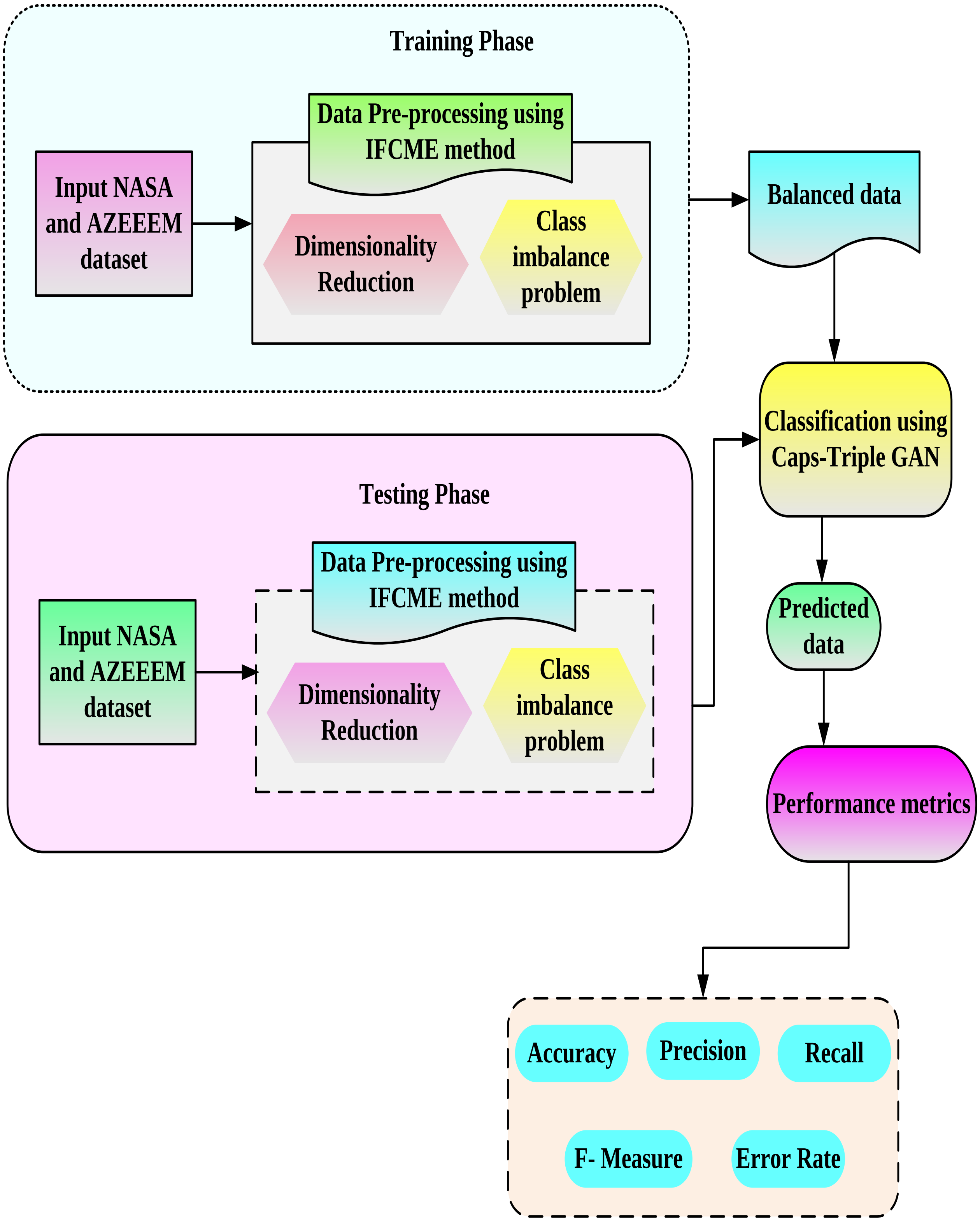

Fig. 1 Block diagram of the proposed method for software prediction using IFCME- Caps-Triple GAN

The training model stages of IFCME are given below:

Stage 1: Stratified random k -fold

Stage 1.1: Datasets are arranged randomly through shuffles of every case and control sample.

Stage 1.2: Here, the dataset is divided into k -subsets. The ratio of cases and controls are calculated, and the k blank cross-validation subsets are generated. The case and control sets are assigned for cross-validation (CV) subsets based on the ratio of case and control.

Stage 2: Generation of training and testing data.

Stage 2.1: Here, single-cross-validation subsets are chosen for the testing data. Then, the extra cross-validation subsets contain training data. The cross-validation subset and testing data are utilized for the test section, and the extra CV subsets are utilized for the train section.

Stage 2.1: Then the set of every available combination is generated. The available combinations are $H = \{H_1, H_2, ...H_P\}$, where H, is represented as the set of available h-locus combinations, h -locus is represented as the h -order epitasis.

Stage 3: Estimation of every possible combination.

Stage 3.1: In this, every h -locus combination is estimated, and multi-factor classes are manufactured based on outcomes with every combination of multi-factor classes in h -locus combination such as $h \geq 2$. Every sample of the training data is assigned to multi-factor classes.

Stage 3.2: Here, the probability of the j^{th} multi-factor classes is derived using the binomial distributions, and its equations are given in equation (1):

$$\begin{cases} m_{ji}* = \dfrac{m_{ji}}{m_{++}} \\ m_{j0}* = \dfrac{m_{j0}}{m_{++}} \end{cases} \tag{1}$$

Where m_{++} is represented as the average number of cases and controls among every multi-factor class m_{ji}, m_{j0} is represented as the cases and controls assure the j^{th} multi-factor classes. The imbalance problem will affect the classifier's oversampling and under-sampling. To avoid this defect, the membership degree function of the FCM method is hired. Under-sampling functions denote the $K(\omega K)$ group, and the lower sampling is denoted as the $M(\omega M)$. Here $m_{j0}*$ is represented as the degree of membership, and its group is represented as the g_j. Then the sampling equations are given in equation (2):

$$\omega_{jx} = \frac{1}{\sum_{k=(k,m)} \left(\dfrac{dis(m_j*, g_x)}{dis(m_j*, g_k)} \right)^{\frac{2}{n-1}}} \tag{2}$$

Where $x = (k, m)$ K the group is defined as the matrix $g_k = \{1,0\}$, and M the group is defined as the matrix $g_m = \{1,0\}$. Then the cross entropy is denoted as the $dis(m_j*, g_x)$ and used to estimate the distance among j^{th} multi-factor classes, results (cases and controls), and its equation is given in equation (3):

$$dis(m_j*, g_x) = \sum g \times \log(m^*) \tag{3}$$

Then the degrees of the sampling groups are derived in equation (4-5):

$$\omega_{jH}* = \begin{cases} 0, & if\ m_{j1}* = 0 \\ \left((1 + \log_{mj0} * m_{j1} *)^{\frac{2}{n-1}} \right)^{-1} & other, \\ 1, & f\ m_{j0}* = 0 \end{cases}, \tag{4}$$

$$\omega_{jL}* = \begin{cases} 0, & if\ m_{j0}* = 0 \\ \left((1 + \log_{mj1} * m_{j0} *)^{\frac{2}{n-1}} \right)^{-1} & other, \\ 1, & f\ m_{j1}* = 0 \end{cases}, \tag{5}$$

From equations (4-5), the dataset imbalance problems are reduced by lowering the sampling groups in the dataset.

Stage 3.3: Here, reducing the dimensionality of four cells such as true positivity (TP), false positivity (FP), false negative (FN), and True Negative (TN) are computed using 2×2 contingency. Then these 4 cells are calculated based on the fuzzy cells given in equation (5-9) with dimensionality reduction based on the multi-factor 3^n with 2×2 dimensionality. Then the dimensionality reduction equations are given in equations (6-9)

$$TP_{fuzzy} = \sum_j m_{j1} * \times \omega_{jK} * \quad (6)$$

$$FP_{fuzzy} = \sum_j m_{j0} * \times \omega_{jK} * \quad (7)$$

$$FN_{fuzzy} = \sum_j m_{j1} * \times \omega_{jM} * \quad (8)$$

$$TN_{fuzzy} = \sum_j m_{j0} * \times \omega_{jM} * \quad (9)$$

For an unbalanced dataset to determine the accuracy, balanced software is predicted using the correct classification rate with the fuzzy model, and its equation is given in equation (10):

$$UBD_{CCRfuzzy} = 0.5\times\left(\frac{TP_{fuzzy}}{TP_{fuzzy}+FN_{fuzzy}}+\frac{TN_{fuzzy}}{FP_{fuzzy}+TN_{fuzzy}}\right) \quad (10)$$

From equation (10), the imbalanced dataset is converted into the balanced dataset, and the dimensionality is reduced. Then to accurately classify the predicted data Caps-TripleGAN classifier is used.

3.3. Capsule triple generative adversarial network (Caps-Triple GAN) for classification

This Caps-TripleGAN classifier accurately calculates the predicted data by reducing the error rate. Here the input data are divided into two phases, training and testing. Training sets are utilized to develop the defect prediction models, and testing sets are utilized to estimate the developed models' efficacy. In this, triple GAN is used to improve the performance of the caps Net.

Then the error rate s decreases in layer 2 of the caps triple GAN neural network, and its output is determined in the Generator FG $F_g(k\,|\,m)$ for reducing the error rate by utilizing the training parameters (k,m) of the Caps Net. In this, the Generator FG is used to detect the software prediction for reducing the dimensionality, and the FG with gradient equation is given in equation (11):

$$AG = \nabla_{AG}\left[\frac{1-\chi}{Ts_D}\sum\nolimits_{(a,b)\approx A_g(a,b)} \log AD(a,b)\right] \quad (11)$$

Then the software detected training parameters for reducing the dimensionality are given to the real distribution system $FC(k,m)$ using the neural network.

Then, the created 3capsules of the triple GAN are represented as a real data pair given as (a_{FC}, b_{FC}) positive training samples of the design parameter. Then the fake data parameter is given as (a_{FD}, b_{FD}) these are known as the negative training samples given to the Discriminator AD. Then, the training process is based on the adversarial process, then the adversarial loss is declared in 3methods, and the equation is given in equation (12):

$$\begin{aligned}\min_{FD,FG}\max_{FC} N(FD,FG,FC) = {} & er_{(k,m)\approx A(k,m)}[\log AD(k,m)] \\ & + \chi er_{(k,m)\approx F(k,m)}[\log 1 - AD(k,m)] \\ & + 1-\chi er_{(k,m)\approx F(k,m)}[\log 1 - AD(m,l)] \\ & + er_{(k,m)\approx F(k,m)}[-\log AD(m\,|\,k)]\end{aligned} \quad (12)$$

Where $N(.)$ implies the perceived value, Then, the trained parameters are utilized, maximizing the possibility of the transmission brand of real and fake data from the Generator as well as the classifier, respectively, $\chi\in(0,1)$ indicating a constant variable parameter to optimize the delay line time of the antenna design with the help of impact of the Generator and the classifier, and $er_{(k,m)\approx F(k,m)}[-\log FD(k\,|\,m)]$ this is known as the error function equation. This way, the software defects are detected, and the dataset dimension and class imbalance problems are reduced using the IFCME-Caps-Triple GAN method.

4. Result and Discussion

This manuscript Improved Fuzzy C-means-based Entropy (IFCME) Approach with CapsNet triple generative adversarial network (Caps-Triple GAN) for predicting software defect imbalanced dataset for reducing dimensionality and Class Imbalance Problem is discussed. Simulation in MATLAB is conducted on PC along with Intel Core i5, 2.50 Giga hetzcentral processing unit, 8GB random access memory, and Windows 7. The evaluation metrics such as precision, recall, F-Measure, accuracy, and specificity are analyzed. The efficiency of the proposed FCME-CAPS-Triple GAN-DR-SDP method is compared with the existing method such as LASSO–SVM-DR-SDP [21], MHCPDP-MAA-DR-SDP [22], and KMFOS-DR-SDP [23] respectively.

4.1. Dataset description

Two datasets are taken to detect the software detection problem from the imbalanced dataset. The datasets are 1. NASA Dataset, these datasets are represented as Soft lab. The soft lab is the Software Research Laboratory at Bogazici University, Istanbul, Turkey. The second dataset is the AEEEM dataset, which contains 61 software metrics: 17 source code metrics, 17 entropy of source code metrics, and 17 churns of source code metrics.

4.2. Performance metrics

Measuring the performance metrics like precision, recall, F-Measure, accuracy, and specificity. For measuring the confusion matrix, it needs True Negative, True Positive, False Negative, and False Positive values.

- True Positive (TP): Predicting the prediction of output for the model and test data
- True Negative (TN): Output predicted by model and output in test data is non-predicted.
- False Positive (FP): When the output predicted by the model is true, then data is non-predicted
- False negative (FN): when the output predicted by the model is false, then data is predicted.

The equations of accuracy, precision, recall, and F-Measure are given in equation (15)

$$Accuracy = \frac{TP + FN}{TP + FP + TN + FN}$$

$$\Pr ecesion = \frac{TP}{TP + TN}$$

$$\operatorname{Re} call = \frac{TP}{TP + FP}$$

$$F - Scorevalue = 2 \times \frac{recall \times precision}{recall + precision}$$

(15)

4.3. Performance analysis using several methods utilized for Software Defect prediction

Here, the performance metrics such as accuracy, precision, recall and f-measure, and an error rate of software defect prediction are analyzed. Then, the efficiency of the proposed FCME-CAPS-Triple GAN-DR-SDP method is compared with the existing method such as LASSO–SVM-DR-SDP [21] MHCPDP-MAA-DR-SDP [22], and KMFOS-DR-SDP [23] respectively.

In this, the software defect is predicted using two datasets, namely NASA and AZEEEM, and the performance of the proposed FCME-CAPS-Triple GAN-DR-SDP method is compared with the existing methods such as LASSO–SVM-DR-SDP, MHCPDP-MAA-DR-SDP, KMFOS-DR-SDP respectively. Figure 2A shows the performance of accuracy. In the NASA dataset, the accuracy of the proposed FCME-CAPS-Triple GAN-DR-SDP method is 34.53%, 29.49%, and 27.85% higher than the existing method such as the former. In the AZEEEM dataset, the performance of the accuracy of the proposed FCME-CAPS-Triple GAN-DR-SDP method is 28.84%, 45.86%, and 23.08% higher than the existing method. Figure 2B shows the performance of precision. In the NASA dataset, the precision of the proposed FCME-CAPS-Triple GAN-DR-SDP method is 32.967%, 19.08%, and 33.97% higher than the existing method. In the AZEEEM dataset, the performance of the precision of the proposed FCME-CAPS-Triple GAN-DR-SDP method is 26.98%, 36.08%, and 47.85% higher than the existing method.

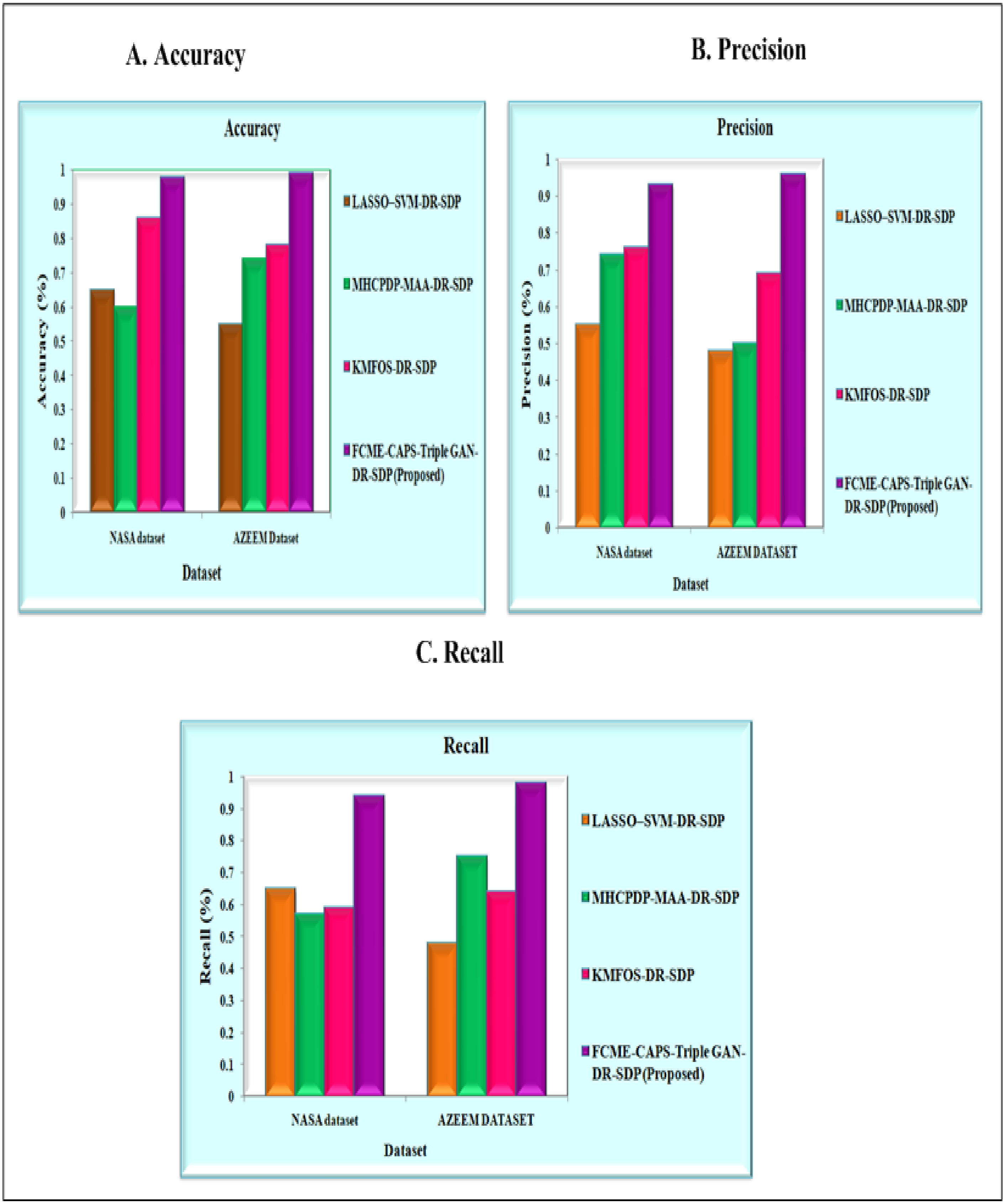

Fig. 2 Accuracy Precision Recall values for SDP datasets

This manuscript Improved Fuzzy C-means-based Entropy (IFCME) Approach with CapsNet triple generative adversarial network (Caps-Triple GAN) for predicting software defect imbalanced dataset for reducing dimensionality and Class Imbalance Problem is successfully implemented. The simulation process is executed in the MATLAB platform. The proposed IFCME- Caps-Triple GAN-DR-SDP attains higher Recall34.86%, 27.97%, 28.74%, High F-Measure35.967%, 56.83%, 19.05%, and the proposed method is compared with the existing methods such as LASSO–SVM-DR-SDP, MHCPDP-MAA-DR-SDP, and KMFOS-DR-SDP respectively.

References

[1] Feng S, Keung J, Yu X, Xiao Y, Bennin K.E, Kabir M.A and Zhang M, “COSTE: Complexity-Based Over Sampling Technique to Alleviate the Class Imbalance Problem in Software Defect Prediction,” *Information and Software Technology,* vol. 129, pp. 106432, 2021.

[2] Ali L, Wajahat I, Golilarz N.A, Keshtkar F and Bukhari S.A.C, “LDA–GA–SVM: Improved Hepatocellular Carcinoma Prediction Through Dimensionality Reduction and Genetically Optimized Support Vector Machine,” *Neural Computing and Applications*, vol. 33, no. 7, pp. 2783-2792, 2021.

[3] Sadiq M.T, Yu X and Yuan Z, “Exploiting Dimensionality Reduction and Neural Network Techniques for the Development of Expert Brain–Computer Interfaces,” *Expert Systems with Applications*, vol. 164, pp. 114031, 2021.

[4] Peng X, Xu D and Chen D, “Robust Distribution-Based Nonnegative Matrix Factorizations for Dimensionality Reduction,” *Information Sciences*, vol. 552, pp. 244-260, 2021.

[5] Sharmila S and Vijayarani S, “Association Rule Mining Using Fuzzy Logic and Whale Optimization Algorithm,” *Soft Computing*, vol. 25, no. 2, pp. 1431-1446, 2021

[6] Sadr A.V, Bassett B.A and Kunz M, “A Flexible Framework for Anomaly Detection Via Dimensionality Reduction,” *Neural Computing and Applications*, pp. 1-11, 2021

[7] Sadr A.V, Bassett B.A and Kunz M, “A Flexible Framework for Anomaly Detection Via Dimensionality Reduction,” *Neural Computing and Applications*, pp. 1-11. 2021

[8] Cheung M, Campbell J.J, Whitby L, Thomas R.J, Braybrook J and Petzing J, “Current Trends in Flow Cytometry Automated Data Analysis Software,” *Cytometry Part A*, 2021.

[9] Maddipati S.S and Srinivas M, “Machine Learning Approach for Classification from Imbalanced Software Defect Data Using PCA & CSANFIS,” *Materials Today: Proceedings*, 2021.

[10] Maddipati S.S and Srinivas M,. Statistical Testing on Prediction of Software Defects,” *EAI Endorsed Transactions on Energy Web*, vol. 5, no. 20, 2018.

[11] Yang C.H, Chuang L.Y and Lin Y.D, “Epistasis Analysis Using an Improved Fuzzy C-Means-Based Entropy Approach,” *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 4, pp. 718-730, 2019.

[12] Wang X, Tan K, Du Q, Chen Y and Du P, “Caps-Triple GAN: GAN-Assisted CapsNet for Hyperspectral Image Classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 7232-7245, 2019.

[13] NASA dataset link, [Online]. Available: https://github.com/klainfo/NASADefectDataset

[14] AZEEEM dataset link, [Online]. Available: https://github.com/bharlow058/AEEEM-and-other-SDP-datasets

[15] Wang K, Liu L, Yuan C and Wang Z, “Software defect prediction model based on LASSO–SVM,” *Neural Computing and Applications*, vol. 33, no. 14, pp. 8249-8259, 2021.

[16] Wu J, Wu Y, Niu N and Zhou M, “MHCPDP: Multi-Source Heterogeneous Cross-Project Defect Prediction Via Multi-Source Transfer Learning and Autoencoder,” *Software Quality Journal*, pp. 1-26, 2021

[17] Gong L, Jiang S and Jiang L, “Tackling Class Imbalance Problem in Software Defect Prediction Through Cluster-Based Over-Sampling with Filtering,” *IEEE Access,* vol. 7, pp. 145725-145737, 2019.

[18] Arvinder Kaur and Kamaldeep Kaur, ”Performance Analysis of Ensemble Learning for Predicting Defects in Open Source Software,” *International Conference on Advances in Computing, Communications and Informatics (ICACCI),* 2014.

[19] Thanh Tung Khuat, My Hanh Le, “Ensemble Learning for Software Fault Prediction Problem with Imbalanced Data,” *International Journal of Electrical and Computer Engineering (IJECE),*vol. 9, no. 4, 2019.

[20] Jaroslaw Hryszko, Lech Madeyski, “Cost Effectiveness of Software Defect Prediction in an Industrial Project,” *Foundations of Computing and Decision Sciences*, vol. 43, no. 1, 2018.

[21] Kazuya Tanaka; Akito Monden; Zeynep Yücel, ”Prediction of Software Defects Using Automated Machine Learning,” 20th *IEEE/ACIS International Conference on Software Engineering*, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2019. Doi: 10.1109/SNPD.2019.8935839.

[22] Pradeep Singh, ”Stacking Based Approach for Prediction of Faulty Modules,” IEEE Conference on Information and Communication Technology (CICT), 2019.

[23] Haitao He, Xu Zhang, Qian Wang, Jiadong Ren, Jiaxin Liu, Xiaolin Zhao, Yongqiang Cheng, ”Ensemble MultiBoost Based on RIPPER Classifier for Prediction of Imbalanced Software Defect Data,” *IEEE Access*, vol. 7.

[24] Z. Li, X. Jing, X. Zhu, and H. Zhang, "Heterogeneous Defect Prediction Through Multiple Kernel Learning and Ensemble Learning," *IEEE International Conference on Software Maintenance and Evolution (ICSME)*, Shanghai, pp. 91-102, 2017. Doi: 10.1109/ICSME.2017.19.

[25] X. Xia, D. Lo, S. J. Pan, N. Nagappan, and X. Wang, "HYDRA: Massively Compositional Model for Cross-Project Defect Prediction," in *IEEE Transactions on Software Engineering*, vol. 42, no. 10, pp. 977-998, 2016. Doi: 10.1109/TSE.2016.2543

[26] Boehm B. W, “Software Engineering Economics,” *Prentice-Hall, Englewood Cliffs,* NJ, 1981.

[27] Satya Srinivas Maddipati, Dr. G Pradeepini, Dr. A Yesubabu, ”Software Defect Prediction using Adaptive Neuro-Fuzzy Inference System,” *International Journal of Applied Engineering Research*, vol. 13, no. 1, pp. 394-397, 2018.

[28] Jyoti Goyal, Bal Kishan, “TLHEL: Two Layer Heterogeneous Ensemble Learning for Prediction of Software Faults,” *International Journal of Engineering Trends and Technology*, vol. 69, no. 4, pp. 16-20, 2021.