Software Defect Prediction using KPCA & CSANFIS

Satya Srinivas Maddipati [a] and Malladi Srinivas [b]

[a] Research Scholar, CSE Department, K L Education Founadations, Vadddeswaram
[b] Professsor, CSE Department, K L Education Founadations, Vadddeswaram

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: Identifying Bugs/Defects in the early stages of software life cycle reduces the effort required in software development. A lot of research has been progressed in predicting software defects using machine learning approaches. In software defect prediction, there are mainly two problems, dimensionality reduction and class Iimbalance. In this paper, we are addressing dimensionality reduction using Kernal Principle Component Analysis and Class Imbalance problem using Cost sensitive Class Imbalance Problem. Kernal Principle Component Analysis transforms non linear high dimensional data into low dimensional space.Cost Sensitive Adaptive Neuro Fuzzy Inference System assigns weights to samples based on class imbalance ratio to alleviate biasing in classification towards majority class. The performance of proposed methodology is measured using Area under ROC Curve (AuC) values. We performed experimentation on Software Defect datasets downloaded from NASA Dataset repository and observed Auc values are increased with our proposed methodology by 5-6%.

Keywords: Software Defect Prediction, Dimensionality reduction, Class imbalance, Kernal Principle Component Analysis, Cost Sensitive Adaptive Neuro Fuzzy Inference System.

1. Introduction

1.1 Defect Prediction

Defect Prediction is the process of identification of Software bugs/defects in the early stages of software development. Defect prediction reduces the cost of software development. By early identification of software defects, Software effort will be reduced. As the software effort reduced, development cost will be reduced. They are two types in defect prediction.

i) Within project defect prediction: In this approach, Defect prediction models are developed for a project and defects in the project are identified using the same models. To develop defect prediction model, the characteristics of the project like Lines of code, Cyclometric complexity measures and Halstead metrics are considered.

ii) Cross project defect prediction: In this approach, Defect prediction models are developed using characteristics of other projects which are used to identify defect proneness in current/developing project. Machine learning models like decision trees, Bayes classifiers, Support vector machines and artificial neural networks can be used for developing software defect prediction models.

1.2Machine learning approaches for defect prediction:

1.2.1 Data Preprocessing:

Data preprocessing techniques in machine learning transforms or reduces the dataset. In software defect prediction, dimensionality reduction will be done by using Principle Component Analysis(PCA). PCA reduces the dataset by projecting high dimensionality data into low dimensional space. PCA suffers from data loss due to non linear characteristics of data. The non linear data can be transformed to low dimensions without data loss using Kernal based Principle Component Analysis(KPCA).

1.2.2 Imbalance Data:

Software defect Prediction dataset is imbalance in nature. Most of the samples are non defective and very less samples are defective. The classifier will be biased due to imbalance nature of data. Data imbalance can be eliminated by using Under sampling or Oversampling. Most of the authors performed research on data imbalance. Synthetic Minority Oversampling is one of the best method to eliminate data imbalance.

1.2.3Supervised Learning:

In supervised learning, the existing dataset is divided into training set and testing set. Training set is used for developing defect prediction models, Testing set is used for evaluating performance of developed model. If the performance metrics are satisfactory, then the models are used for prediction of defects in unseen/new projects(developing projects). Supervised learning techniques named J48, Random forests and Naïve Bayes classifiers are used as leaning classifiers for defect prediction and these classifiers are evaluated using Precision, Auc and Mean absolute error etc.(A. Chug , 2013). Figure 1 shows methodology for Supervised learning

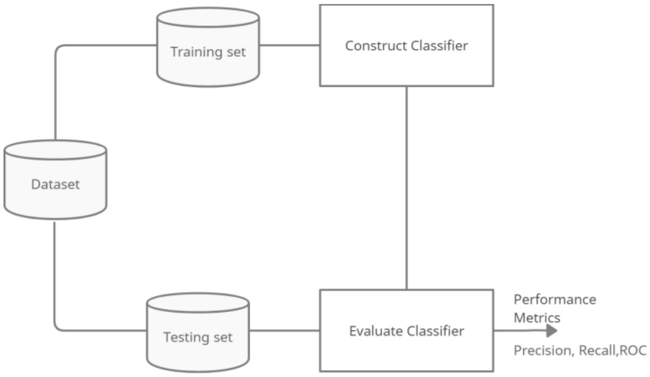


Figure 1. Supervised learning

1.2.4 Unsupervised learning:

In Unsupervised learning, the dataset is segmented into two clusters defective and non defective. Un supervised leaning techniques like K-means, Hierarchical clustering and Density based clustering are applied on defect dataset and these clustering techniques are evaluated based on Sum of Square Error(SSE). The algorithm with least SSE value is considered as best clustering algorithm. Figure 2 shows methodology for Unsupervised learning.

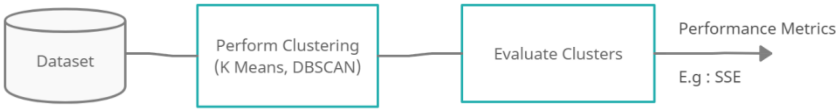


Figure 2. Unsupervised learning

2. Literature Review

(Ye Xia et. al 2013)surveyed the usage of dimensionality reduction techniques for data preprocessing in estimation of software defects. They applied three base classifiers Decision trees , Naïve bayes and support vector machines on NASA software defect dataset repository and concluded that with the proposed approach less than 10 metrics gets better performance in Software defect identification. Ruchika Malhotra et. al studied various feature extraction techniques such as Linear discriminant analysis, Principle Component analysis, Kernal based principle component analysis and auto encoders in Software defect prediction. They applied Support Vector Machine as base classifier and concluded that Auto encoders achieves better performance in dimensionality reduction compared with other approaches (R. Malhotra 2020). (Y.Gao et. al 2017) proposed Geometric mean for subspace learning and conditional random field for prediction of software defects. Geometric mean for sub space learning chooses best set of features from original dataset. (H.Wei et. al 2018) proposed Neighborhood preserving embedded support vector machine for software defect identification and concluded that the proposed methodology improves F-score by 3%- 4%. Neighborhood Preserving Embedded algorithm preserves the local structure of data distribution. (S.Ghosh et. al 2018) proposed Non linear manifold detection approach for optimization of feature selection in software defect prediction.

They applied Decision Trees and Random forest as base classifiers and evaluated the results statistically by Friedman test followed by Wilcoxon Sign Rank test and proves the proposed methodology improves the accuracy in software defect prediction.

(H. Lu et. Al 2014) proposed feature compression technique especially multi dimensionality scaling and Random forest as base classifier. They concluded that the performance of classifier improved with active learning . Active learning dynamically adapts new knowledge and improves predictive performance. (Haijin Ji et 2017) proposed maximal information coefficient and automatic clustering for attribute selection in software defect dataset. This approach initially computes maximal information coefficient matrix between all attributes and spectral clusters will be formed based on maximal information coefficients. Optimal number of clusters are selected by Calinski-Harabasz criterion and best set of attributes are selected. (Y. Zhou, 2019) proposed Kernal based Principle Component analysis as a dimensionality reduction technique and used support vector machines as base classifier for predicting software defects and concluded that the proposed methodology gets better precision value. Non characteristics of software defect dataset can be modeled effectively using non linear kernel in support vector machines.

(K. E. Bennin et. al 2018) proposed an efficient and novel synthetic oversampling technique, called MAHAKIL to overcome from class Imbalance problem in software defect prediction. This method constructs new instances by inheriting traits from parent classes and achieves diversity in data distribution. They proved that their proposed method improves PF values compared to other class imbalance techniques. (X. Jing et. al 2017) proposed subclass discriminant analysis for feature selection in Within Project class imbalance problem and Semi supervised transfer Component analysis with improved subclass discriminant analysis for feature selection in Cross Project class imbalance problem. They concluded that their proposed approach gets better accuracy compared to traditional methods. (L. Gong et. al 2020) proposed stratification embedded in nearest neighbor for balancing data distribution to alleviate class imbalance problem in software defect prediction. For Within project class imbalance problem, they directly transferred stratification embedded in nearest neighbor. For cross project class imbalance problem, they transferred component analysis. They applied ensemble learning with weighted votes and proved that AuC, Recall, Probability of false positives and F measure values are improved with proposed methodology. Software defects at software change level can be predicted by suing Just in Time Software Defect prediction(JIT-SDP).

JIT-SDP suffers from class imbalance problem.G.G. (Cabral et. al 2019) proposed a novel class imbalance technique that overemphasizes one class over other class. Ankush Joon et. al combined noise removal, Class imbalance distribution and feature selection techniques to optimize feature selection and proved that the proposed methodology improves accuracy, precision, recall, F-measure and Auc values. L. Gong et. al investigated impact of class overlap in software defect prediction and also proposed Improved K-means Clustering Cleaning approach for class overlapping and class imbalance problems in Software defect prediction and concluded that the proposed approach improves Recall and AuC values in Within Project Defect Prediction and Cross Project Defect Prediction. S. Huda et. al proposed ensemble oversampling technique to alleviate class imbalance problem in Software defect prediction. Ensemble oversampling technique greatly reduces false negative rate and predicts the faulty modules with high accuracy. Hence this techniques reduces expensive cost of software development by more accurately predicting faulty modules in advance. T. Chakraborty et. al proposed an hybrid approach, Hellinger net, a deep feed forward neural network with built in hierarchy that maps a tree to network model to utilize strength of skewness in insensitive proximity measure. They conducted the experiments on NASA dataset repository.

Q. Zha et. al proposed adaptive centre-weighted oversampling to address class imbalance problem in Software defect prediction. In this method, synthetic minority samples are generated by neighours in each minority class sample with in neighborhood range later oversampling will be performed by using weights assigned to minority class samples. L. Gong et. al proposed noise filtering in cluster based oversampling to address class imbalance and noise removal problems in software defect prediction and concluded that Recall values are improved with their proposed approach compared to SMOTE and Borderline SMOTE, K means SMOTE and ADASYN. R. B. Bahaweres et. al combined Neural network and SMOTE with the hyper parameters are optimized using random search to alleviate class imbalance problem in Software defect predictionand proved that the recall value increases by 45.99% with their proposed approach. Y. Liu et. al proposed Extended nearest neighbor algorithm for data cleaning, SMOTE algorithm for class imbalance and Simulated Annealing algorithm to optimize four layer back propagation neural network for classification. S. S Maddipati et. al proposed Cost sensitive Adaptive Neuro fuzzzy inference system for overcoming class imbalance problem in software defect prediction. Data distribution among different software projects will be varied greately. Cong Jin et. al proposed kernel twin support vector machines based on domain adaptation for cross project defect prediction. The parameters are optimized using Particle Swarm Optimization technique and concluded that their proposed methodology gets same or better results when the training data is in sufficient. Amirabbas Majd et. al proposed a novel approach,SLDeep, for software defect prediction using statement level metrics. They defined 32 statement level metrics such as operands and operators used in the statement and applied Long Short term memory as a learning model. They showd the Proposed meethodology recall, precision and accuracy values as 0.98,0.57 and 0.7 respectively. S.S Maddipati et. al proposed ensemble learning approach for cost effective learning of software defect prediction. K. Zhu et. al proposed Improved Transfer technique in Naïve Bayes algorithm for Within Project defect prediction and Cross Project defect prediction. In this technique, samples are weighted with dimensionality weight and data gravity.

In previous surveys ,some of the researchers addressed dimensionality reduction and other researchers addressed class imbalance problem in Software defects. To improve accuracy in prediction of software defects both the problems i.e, Dimensionality reduction and class imbalance problem must be addressed

3. Methodology

In our research work, We are addressing both dimensionality reduction and data imbalance problem in Software Defect Prediction. We are applying KPCA for dimensionality reduction and Cost Sensitive Adaptive Neuro Fuzzy Inference System for imbalance classification. These Cost metric values are derived from imbalance ratio.

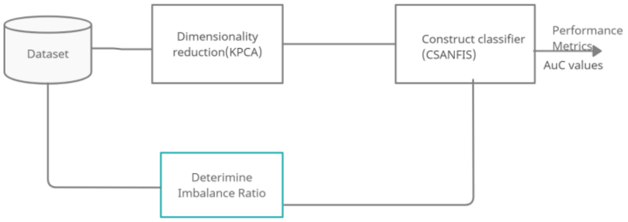

Dataset
Dimensionality reduction(KPCA)
Construct classifier (CSANFIS)
Performance Metrics
AuC values
Deterimine Imbalance Ratio

Figure 3. Proposed Methodology

3.1 Algorithm:

```
//K fold Cross validation
1  .Dataset<-read.csv("pc1.csv")
2.  Trainset<-sample(Dataset,0.9)
3.  Testset<-sample(Dataset,0.1)
//Determining Imbalance Ratio
4.  IBR<-nrow(Dataset$defect=FALSE)/nrow(Dataset$defect=TRUE)
// Dimensionality Reduction using PCA
5. data_transform<-kpca(Trainset,kernel="rdfdot")
//Construct classifier
6 .Model<-ANFIS(data_transform,max.iter=10,stepsize=0.01,type.tnorm="MIN",type.snorm="MAX",weight=IBR)
//Evaluating Classifier
7. target<-predict(Model,Testset)
8. roc_obj<-roc(Testset$defect,target)
9. auc_val<-auc(roc_obj)
```

4. Result

The In this research work, we applied Kernal based Principle Component Analysis on PC1 dataset and projected new features in low dimensional data space. Figure 4 shows Principle Components for software defect prediction.

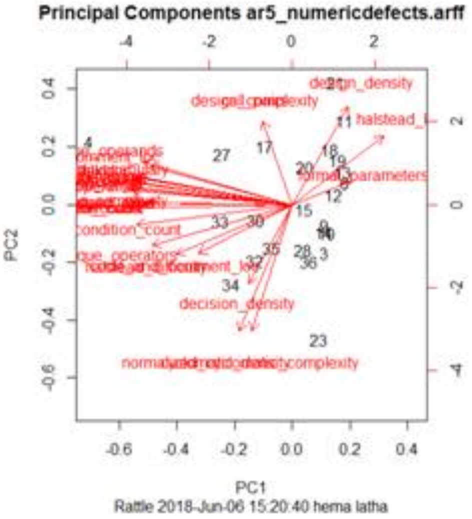
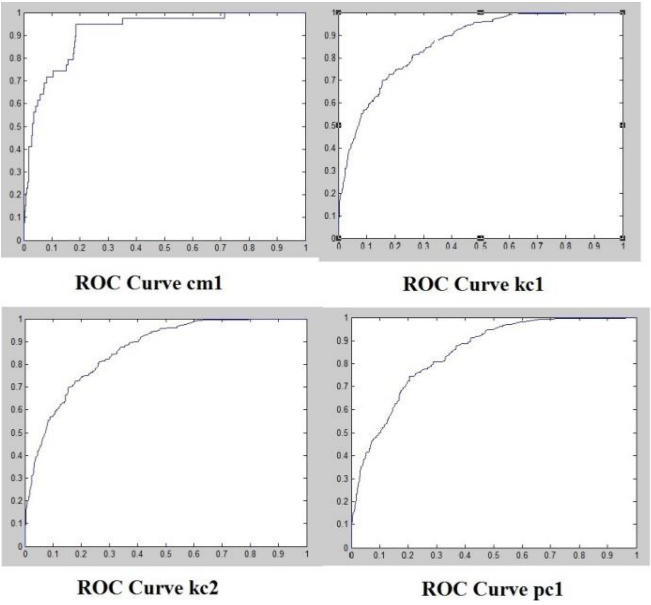


Figure 4. Kernal Principle Component Analysis

Cost Sensitive Adaptive NEuro Fuzzy Inference System was modeled using low dimensional data. The output from ACSANFIS model was Sugeno Fuzzy Inference System. This Sugeno Fuzzy Inference System was evaluated using test data. Confusion matric was constructed. ROC curves were generated from confusion matrix. Area under ROC curves was determined. Fig 5 shows ROC curves generated on SDP dataset



ROC Curve cm1

ROC Curve kc1

ROC Curve kc2

ROC Curve pc1

Figure 5. ROC curves generated by CSANFIS

5. Conclusion

In software development life cycle, early detection of software defects plays a major role. Various classification techniques were proposed for software defect prediction. In this research work, we applied Kernal Principle Component Analysis as dimensionality reduction, CSANFIS as base classifier. This model improves AuC values in prediction of Software defects

References

1. Amirabbas Majd, Mojtaba Vahidi-Asl, Alireza Khalilian, Pooria Poorsarvi-Tehrani, Hassan Haghighi,SLDeep(2020): *Statement-level software defect prediction using deep-learning model on static code features,Expert Systems with Applications, 147,113156,ISSN 0957-4174,https://doi.org/10.1016/j.eswa.2019.113156.*
2. Aroulanandam, V.V., Latchoumi, T.P., Bhavya, B., Sultana, S.S. (2019). Object detection in convolution neural networks using iterative refinements. Revue d'Intelligence Artificielle, Vol. 33, No. 5, pp. 367-372. https://doi.org/10.18280/ria.330506
3. Bahaweres .R .B, F. Agustian, I. Hermadi, A. I. Suroso and Y. Arkeman, (2020)"*Software Defect Prediction Using Neural Network Based SMOTE," 2020 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI), Yogyakarta, Indonesia, pp. 71-76, doi: 10.23919/EECSI50503.2020.9251874.*
4. Balamurugan, K., Uthayakumar, M., Ramakrishna, M. and Pillai, U.T.S., 2020. Air jet Erosion studies on mg/SiC composite. Silicon, 12(2), pp.413-423.
5. Balamurugan, K., 2020. Compressive Property Examination on Poly Lactic Acid-Copper Composite Filament in Fused Deposition Model–A Green Manufacturing Process. Journal of Green Engineering, 10, pp.843-852.
6. Bennin .K .E, J. Keung, P. Phannachitta, A. Monden and S. Mensah, (2018). Balamurugan, K., 2020. Compressive Property Examination on Poly Lactic Acid-Copper Composite Filament in Fused Deposition Model–A Green Manufacturing Process. Journal of Green Engineering, 10, pp.843-852.*699-699, doi: 10.1145/3180155.3182520.*
7. Bhasha, A.C. and Balamurugan, K., 2020, July. Multi-objective optimization of high-speed end milling on Al6061/3% RHA/6% TiC reinforced hybrid composite using Taguchi coupled GRA. In 2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE) (pp. 1-6). IEEE.
8. Cabral .G .G, L. L. Minku, E. Shihab and S. Mujahid, (2019) *"Class Imbalance Evolution and Verification Latency in Just-in-Time Software Defect Prediction," 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), Montreal, QC, Canada, pp. 666-676, doi: 10.1109/ICSE.2019.00076.*
9. Chakraborty .T and A. K. Chakraborty, *"Hellinger Net: A Hybrid Imbalance Learning Model to Improve Software Defect Prediction," in IEEE Transactions on Reliability, doi: 10.1109/TR.2020.3020238.*
10. Chug .A and S. Dhall, (2013) *"Software defect prediction using supervised learning algorithm and unsupervised learning algorithm," Confluence 2013: The Next Generation Information Technology Summit (4th International Conference), Noida, pp. 173-179, doi: 10.1049/cp.2013.2313.*
11. Cong Jin, (2021)Cross-*project software defect prediction based on domain adaptation learning and optimization, Expert Systems with Applications, 171, 114637, ISSN 0957-4174,https://doi.org/10.1016/j.eswa.2021.114637.*
12. Garikipati P., Balamurugan K. (2021) Abrasive Water Jet Machining Studies on $AlSi_7$+63%SiC Hybrid Composite. In: Arockiarajan A., Duraiselvam M., Raju R. (eds) Advances in Industrial Automation and Smart Manufacturing. Lecture Notes in Mechanical Engineering. Springer, Singapore. https://doi.org/10.1007/978-981-15-4739-3_66
13. Gao .Y, C. Yang and L. Liang, (2017) *"Software defect prediction based on geometric mean for subspace learning," 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, pp. 225-229, doi: 10.1109/IAEAC.2017.8054011..*
14. Gong .L, S. Jiang, L. Bo, L. Jiang and J. Qian , (2020). *"A Novel Class-Imbalance Learning Approach for Both Within-Project and Cross-Project Defect Prediction," in IEEE Transactions on Reliability, 69(1), pp. 40-54, March 2020, doi: 10.1109/TR.2019.2895462.*
15. Gong .L, S. Jiang, R. Wang and L. Jiang, (2019) *"Empirical Evaluation of the Impact of Class Overlap on Software Defect Prediction," 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), San Diego, CA, USA, pp. 698-709, doi: 10.1109/ASE.2019.00071.*
16. Gong .L, S. Jiang and L. Jiang, (2019*) "Tackling Class Imbalance Problem in Software Defect Prediction Through Cluster-Based Over-Sampling With Filtering," in IEEE Access, 7, pp. 145725-145737, doi: 10.1109/ACCESS.2019.2945858.*

17. Ghosh .S, A. Rana and V. Kansal, (2018). *"A Hybrid Nonlinear Manifold Detection Approach for Software Defect Prediction," 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, pp. 453-459, doi: 10.1109/ICRITO.2018.8748788.*
18. Gowthaman, S., Balamurugan, K., Kumar, P.M., Ali, S.A., Kumar, K.M. and Gopal, N.V.R., 2018. Electrical discharge machining studies on monel-super alloy. Procedia Manufacturing, 20, pp.386-391.
19. Huda .S et al.,(2018) *"An Ensemble Oversampling Model for Class Imbalance Problem in Software Defect Prediction," in IEEE Access, 6, pp. 24184-24195, doi: 10.1109/ACCESS.2018.2817572.*
20. Ji .H, S. Huang, Y. Wu, Z. Hui and X. Lv, (2017*) "A New Attribute Selection Method Based on Maximal Information Coefficient and Automatic Clustering," International Conference on Dependable Systems and Their Applications (DSA), Beijing, China, 2017, pp. 22-28, doi: 10.1109/DSA.2017.13*
21. Jing .X, F. Wu, X. Dong and B. Xu, (2017) *"An Improved SDA Based Defect Prediction Framework for Both Within-Project and Cross-Project Class-Imbalance Problems," in IEEE Transactions on Software Engineering, 43(4), pp. 321-339, doi: 10.1109/TSE.2016.2597849.*
22. Joon .A, R. Kumar Tyagi and K. Kumar, (2020) *"Noise Filtering and Imbalance Class Distribution Removal for Optimizing Software Fault Prediction using Best Software Metrics Suite," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 1381-1389, doi: 10.1109/ICCES48766.2020.9137899.Mustacoglu, A. F, Catak, F. O., & Fox, G. C. Password-based encryption approach for securing sensitive data. Security and Privacy, 3(5).*
23. Lu .H, E. Kocaguneli and B. Cukic, (2014) *"Defect Prediction between Software Versions with Active Learning and Dimensionality Reduction," IEEE 25th International Symposium on Software Reliability Engineering, Naples, Italy,, pp. 312-322, doi: 10.1109/ISSRE.2014.35.*
24. Liu .Y, F. Sun, J. Yang and D. Zhou,(2020) *"Software Defect Prediction Model Based on Improved BP Neural Network," 2019 6th International Conference on Dependable Systems and Their Applications (DSA), Harbin, China, pp. 521-522, doi: 10.1109/DSA.2019.00095*
25. Malhotra .R and K. Khan, (2020*) "A Study on Software Defect Prediction using Feature Extraction Techniques," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, pp. 1139-1144, doi: 10.1109/ICRITO48877.2020.9197999.Wei .H, C. Shan, C. Hu, H. Sun and M. Lei, (2018) "Software defect distribution prediction model based on NPE-SVM," in China Communications, 15(5), pp. 173-182, doi: 10.1109/CC.2018.8387996.*
26. Ranjeeth, S., Latchoumi, T.P., Sivaram, M., Jayanthiladevi, A. and Kumar, T.S., 2019, December. Predicting Student Performance with ANNQ3H: A Case Study in Secondary Education. In 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE) (pp. 603-607). IEEE.
27. Satya Srinivas Maddipati, Malladi Srinivas,(2021) *Machine learning approach for classification from imbalanced software defect data using PCA & CSANFIS,Materials Today: Proceedings,ISSN 2214-7853, https://doi.org/10.1016/j.matpr.2021.01.471.*
28. Satya Srinivas Maddipati and Malladi Srinivas, (2021) *"An Hybrid Approach for Cost Effective Prediction of Software Defects" International Journal of Advanced Computer Science and Applications(IJACSA), 12(2), http://dx.doi.org/10.14569/IJACSA.2021.0120219*
29. Xia .Y, G. Yan and Q. Si, (2013*). "A Study on the Significance of Software Metrics in Defect Prediction," Sixth International Symposium on Computational Intelligence and Design, Hangzhou, China, pp. 343-346, doi: 10.1109/ISCID.2013.199.*
30. Yookesh, T.L., Boobalan, E.D. and Latchoumi, T.P., 2020, March. Variational Iteration Method to Deal with Time Delay Differential Equations under Uncertainty Conditions. In 2020 International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 252-256). IEEE.
31. Zha .Q, X. Yan and Y. Zhou,(2018) *"Adaptive Centre-Weighted Oversampling for Class Imbalance in Software Defect Prediction," 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), Melbourne, Australia, pp. 223-230, doi: 10.1109/BDCloud.2018.00044.*
32. Zhu .K, N. Zhang, S. Ying and X. Wang, (2020) *"Within-project and cross-project software defect prediction based on improved transfer naive bayes algorithm," Computers, Materials & Continua, 63(2), pp. 891–910.*
33. Zhou .Y, C. Shan, S. Sun, S. Wei and S. Zhang, (2019) *"Software Defect Prediction Model Based On KPCA-SVM," IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable*

Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, pp. 1326-1332, doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00244.