International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Solving the Imbalanced Class Problem in Software Defect Prediction using GANS

S.Kaliraj, Aman Jaiswal

Abstract*: Prediction of software defects is a highly researched and important domain for cost - saving advantage in software development. Different methods of classification using attributes of static code were used to predict defects in software.However, the defective instances count is very minimal compared to the count of non - defective instances and this leads to imbalanced data, where the ratio of data class is not equal. For such data, conventional machine learning techniques give poor results.While there are different strategies to address this issue, normal oversampling methods are different versions of the SMOTE algorithm, These approaches are based on local information,instead of the complete distribution of minority class.GANs is used to approximate the true data distribution of minority class data used for software defect prediction.

***Index Terms*: Class imbalanced, Software defects , Gans.**

I. INTRODUCTION

The available data for current software defect prediction is imbalanced, the project's goal is to resolve the problem of imbalanced class distribution in the dataset available for the early fault prediction model based on machine learning.Prediction of software defects is a highly researched and important domain for cost - saving advantage in software development. Different methods of classification using attributes of static code were used to predict defects. However, the defective instances count is very minimal compared to the count of non - defected instances and this leads to imbalanced data. For such data, conventional machine learning techniques give poor results.While there are different strategies to address this issue, normal oversampling methods are different versions of the SMOTE algorithm, These approaches are based on local information,instead of the complete distribution of minority class.GANs is used to approximate the true data distribution of minority class of data used for early software defect prediction.The data thus generated is compared with the data generated from other techniques like random-oversampling,smote and adasyn.

Manuscript published on 30 September 2019
* Correspondence Author
S.Kaliraj*, Department of Software Engineering, Kattankulathur Campus, SRM Institute of Science and Technology (formerly known as SRM University), India.
Aman Jaiswal, Department of Software Engineering, Kattankulathur Campus, SRM Institute of Science and Technology (formerly known as SRM University), India.

mainly due to the lack of data cleaning and removal of

II. LITERATURE SURVEY

Bruce Christianson, David Bowes, Yi Sun,

Neil Davey and David Gray published the result that Software prediction experiments conducted with the help of NASA's Metrics Data Program may have had erroneous findings[20]. This is Repeated data points in training and testing data published by the IEEE Society in April 2011. The inference that can be made of this is that the NASA dataset requires a lot of pre - processing before machine learning can be used. Another literature survey led us to the finding that Imbalanced data is very commonly found in the real world software data and must be taken into account when predicting defects. This was published in Model Software Defects as Anomalies in October 2017[13].It shows that the majority of data available for fault prediction is imbalanced .ie the minority class instances is very less than the majority class instances and this creates an imbalanced dataset for training a model. The survey of another paper that deals with generating data using Generative Adversarial Networks gave us the results that shows that cGan can perform better than other forms of classifiers.It can be concluded that The cGAN extension of the GAN framework can be used for minority class oversampling[17]. The survey of another paper that deals with an improved approach for SDA based defect prediction gave us the overview that class-imbalance heavily affects the performance of prediction[9].

III. SURVEYED TECHNIQUES

RANDOM OVER SAMPLING:
This technique randomly selects data points from the original data set and then replicates the data points that represents the minority data points.

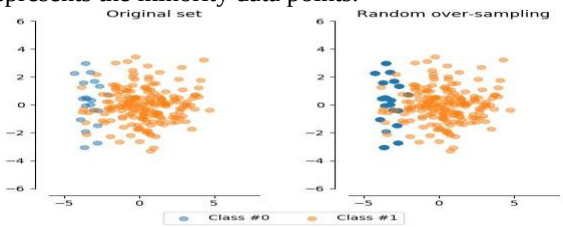

Fig.1 Random over sampling.

The replicated data points are then copied into the training dataset. The fig.1 shows an image thatshows minority class represented by blue dots being replicated.Random over-sampling is implemented in our project using imblearn python library for
rapid development and testing.
SMOTE: Synthetic Minority Over-sampling Technique
In this technique, the majority class is not taken into account instead of the minority class is taken into consideration. The synthetic dataset is created by taking each minority point and k nearest elements and a synthetic element is created at the halfway of the instance and its nearest neighbor.

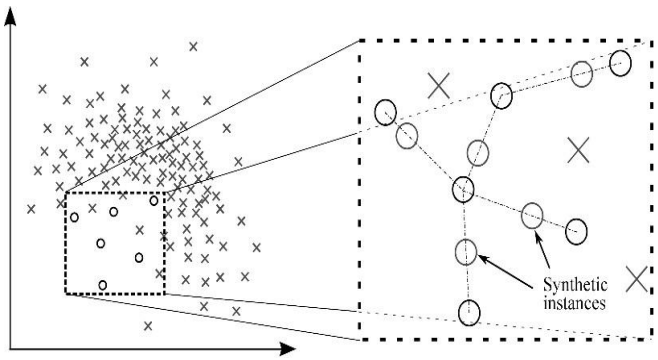

Fig.2 SMOTE.

Smote is able to generate synthetic data in between the line that represents the minority class samples.Figure 2 shows synthetic instances being inserted in the intermediate space between the line joining two minority points. the imblearn library takes care of any noisy samples that may be generated due to unclear separation between majority and minority class clusters.this is done by border-lining the instances of both the classes and using this border to distinguish between the majority and minority classes.
ADASYN: It allows to generate adaptively the minority samples according to their distribution.The amount of synthetic data generated is more for the minority instances that are harder to learn and less instances are generated for data points that are easier to learn like majority class instances.

IV. PROPOSED TECHNIQUES

Why GAN?
Generative adversarial networks is relatively a new technology than compared to other forms of oversampling methods mentioned in the paper.The majority of application of Gans have been limited to the field of computer vision and other image related applications.It has been used to generate image samples.The training data for such models have been from the image domain and also the output have been from the image domain.An effectively trained Gan is able to generate hyper-realistic images that have never existed before. The aim of the paper is to exploit this property of the generative adversarial neural networks and create realistic data for minority class and thus use this data for training a prediction model.
GANS: The paper aims to evaluate the generators ability to effectively generate synthetic datasets for the minority class instances.The structure of Gans can be seen in the figure 3.It consists of two separate networks commonly Called as generator and discriminator.The job of the generator is to generate artificial samples and the job of discriminator is to identify with a good enough accuracy that the generated samples are real or fake.A network is said to be trained if it becomes difficult for the discriminator to tell the difference between real and generated data. ie. the probability according to discriminator of the data being fake or real comes close to 0.50 .The process of training the network is initiated by generating a random noise from latent space to serve as an input to the generator. The generator then generates an output according to the structure and weights of the generator.The project deals with tabular data rather than image data which is common for gans. The output is randomly inputed into the discriminator therefore the discriminator is unaware if the inputed data is generated data or real data.This process can be seen in the form of a switch in fig(3) before the discriminator.The discriminator then output the probability of the inputed data being real.This is done by setting the output layer of the discriminator network as a sigmoidal function. The output of the discriminator is simultaneously used by the generator for modification of weights and discriminator for fine-tuning. This process goes on until the discriminator is unable to determine if the inputed data is real or fake. in our case the data is our processed data ie. 2 PCA components and a label.

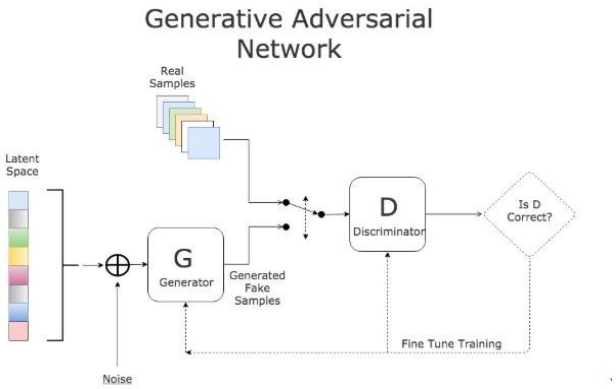

Fig.3

GAN EVALUATION
CRITERIA :
The Gans performance to be used as an oversampler is evaluated and then matched that of with Random oversampling , Smote and Adasyn. Total accuracy is not an appropriate way to measure accuracy for imbalanced dataset therefore we use f- measure to compare different techniques.

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

F-Measure is a good way to measure the accuracy of a model trained on imbalanced dataset.It takes into account both the precision and recall to gather the result.

V. MODELS

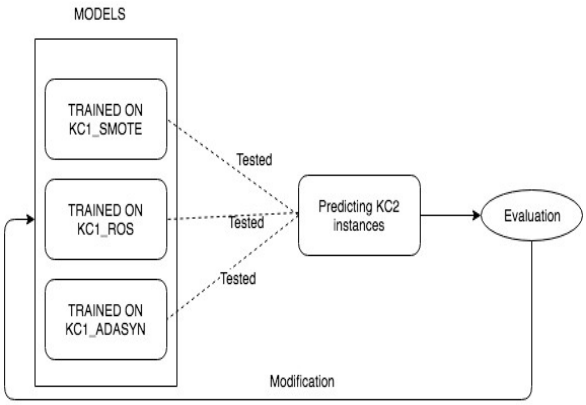

Fig.4 Integration.

The above figure(4) describes the integration testing process in our paper .The model is trained from different dataset available after applying each oversampling method separately. These models are then used to predict the defective instances in unseen data and then compared.

The Prediction model for identifying the defective instances in the MDP dataset is a simple 3 layer neural network with one input layer , one hidden layer and one output layer.The original dataset contains 22 features.Considering the complete set of 22 features is very resource intensive and may not always lead to a better result, Therefore the 22 features are reduced to 2 features with the help of principal component analysis.

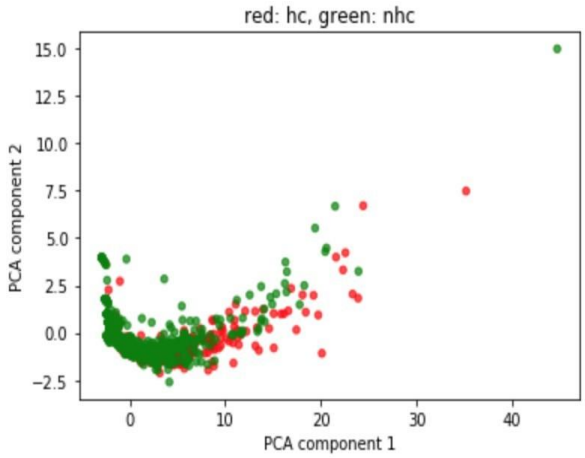

Fig.5 PCA.

The PCA algorithm is able to preserve the correlation between the features and also reduce its dimensionality. Fig(5) shows the relation between PCA component 1 and PCA component 2. The 2 components along with the target values (true,false) is used as input for training the model.PCA is done after over-sampling using Smote, Random over-sampling and ADASYN. Pre-existing libraries are used to make the over-sampling process efficient and easy. As can be seen in fig(4) Three files are generated after over-sampling , one for each technique.These file are used for training the neural network in a parallel workflow.

A neural network is trained on the PCA components with adam optimiser and binary cross entropy as the loss function.Number of epochs used varies between 15-20 for each over-sampling method. The weights of the network are updated using back-propagation.

VI. RESULT

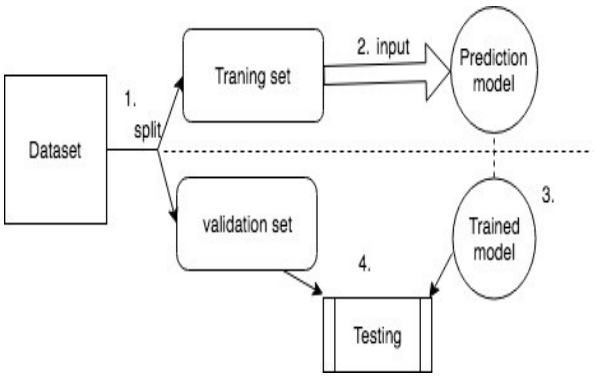

Fig.6 Data splitting.

As seen in fig(6) The dataset is split into two sets training set and validation set. The ratio of the sets is kept as .10 for improving the training process as much possible. The training set is used to train the model and the validation set is used for validation. The validation set is kept completely isolated during the training process. After the model is trained the validation set is used to calculate the performance of the model which in our case is the F-measure for comparing all the over -sampling techniques. The accuracy and loss is plotted for Smote , Random oversampling and GANs. The formulas used for result evaluation is given in fig(7).

$$\text{Accuracy} = \frac{tp+tn}{tp+tn+fp+fn} \qquad \text{Precision} = \frac{tp}{tp+fp}$$

$$\text{Recall} = \frac{tp}{tp+fn} \qquad \text{Recall} = \frac{tp}{tp+fn}$$

Fig.7

The evaluation process must be without bias hence only kc1 dataset is used to evaluate the performance of each method.Confusion matrix is calculated for each technique. fig(8) shows the confusion matrix structure.

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Fig.8 confusion matrix.

The confusion matrix shown in fig(8) is used to show the number of true positives,false positives, false negatives and true negative.

Table.1 Confusion Matrix for various algorithms.

RANDOM OVERSAMPLING		
	Predicted	Predicted
Actual	Non-Defective	Defective
Non-Defective	296	119
Defective	19	88
SMOTE		
	Predicted	Predicted
Actual	Non-Defective	Defective
Non-Defective	303	112
Defective	19	88
ADASYN		
	Predicted	Predicted
Actual	Non-Defective	Defective
Non-Defective	262	153
Defective	14	93
GANS		
	Predicted	Predicted
Actual	Non-Defective	Defective
Non-Defective	263	152
Defective	14	93

Table.1 shows the confusion matrix for the tried oversampling techniques.Confusion matrix shows the predicted vs actual values in a matrix format.

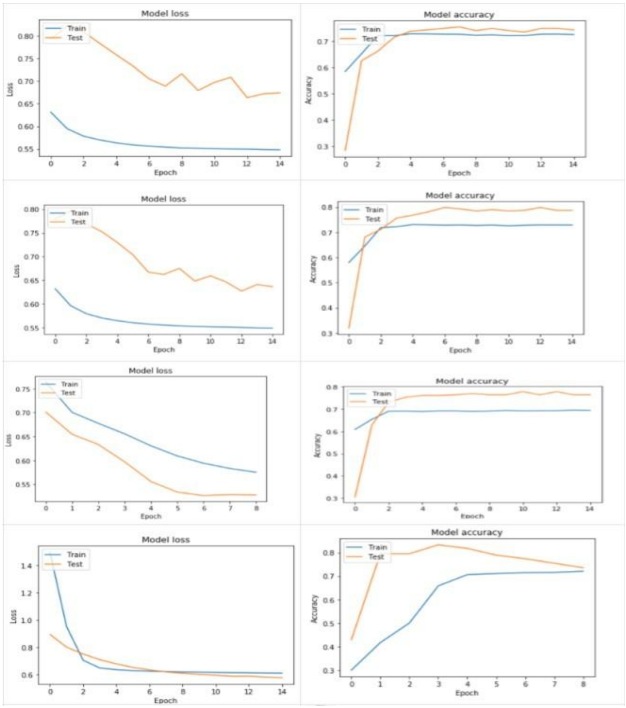

Fig.9 Loss and model accuracy for oversampling algorithms.

Fig(9) shows the loss and model accuracy for different Over sampling algorithms .The Gans over sampling algorithm shows a slight better graph than other algorithms. The f-measure is calculated for each matrix using the confusion matrix.

Table.2 Precision, Recall, F1-score and Support

	Precision	Recall	F1-score	Support
Non-Defective	0.93	0.71	0.81	415
Defective	0.43	0.82	0.56	107
Non-Defective	0.94	0.73	0.82	415
Defective	0.44	0.82	0.57	107
Non-Defective	0.95	0.63	0.76	415
Defective	0.38	0.87	0.53	107
Non-Defective	0.95	0.63	0.76	415
Defective	0.38	0.87	0.53	107

The results are derived from the confusion matrix by the calculation of f- measure score shown in table(2).Results can be visualized through the loss and accuracy graph.The x -axis of loss graph shows the number of epochs and y axis - shows amount the loss.In the accuracy graph the y-axis shows the accuracy.

The Smote methods shows the highest f1- measure initially, though the Gans method is able to produce high quality and correlated data. It is seen that for the over sampling NASA MDP data set, the Random over sampler method works better than ADASYN.

VII. CONCLUSION AND FUTURE WORK

The results show that GAN has the ability to perform better compared to the other methods if modeled and trained properly.The improvement in the results is due to the ability of the gan to model the true distribution of training data if given enough training time and resources.This is in contrast to the other oversampling methods where adaptive method is applied to generate minority class instances in the input space of safe areas.Training Gans require a lot of effort and time compared to other methods but once the training process is complete generation of synthetic instances is rapid and fast .The various other types of gan also be used for oversampling like WGAN.In this paper we dealt with tabular data which is not traditional for training Gans, some method can be found to transfer this data in the image domain and then train the image data with Gan, This method may provide interesting results .Future work can be done by altering the structure of generator or discriminator,which can improve the results.Implementation of the WGAN may give the paper's suggested better results. Newer models can be made that takes into consideration that the training data for minority instances is generated by gans ,

which can be used to optimize the prediction model which takes such training data.

REFERENCES

1. Akbani, R., Kwek, S., & Japkowicz, N. (2004). Applying support vector machine to im-balanced datasets. Machine Learning: ECML
2. Barua, S., Islam, Md. M., Yao, X., & Murase, K. (2014). MWMOTE – weighted majority minority over-sampling technique for imbalanced data-set learning.
3. Batista, G., Prati, R., & Monard, M. (2004). A Balancing Machine Learning Training Data Be-havior Study of Several Methods. ACM SIGKDD Explorations Newsletter-Special problem on unbalanced dataset learning
4. Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2012). DBSMOTE: Densi- ty-based synthetic minority over-sampling technique. Applied Intelligence
5. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Syn- thetic minority over-sampling technique. Journal of Artificial Intelligence Research
6. Chawla, N. V., Japkowicz, N., & Kolcz, A. (2003). Workshop learning from im-balanced data sets II. In Proceedings of inter-national conference on machine learning.
7. Chawla, N. V. (2005). Data mining for imbalanced data sets: an overview. In Data mining and knowledge discovery handbook.
8. Chawla, N. V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2003). SMOTEBoost: Im- proving prediction of minority class in boosting. In Proceedings of seventh European conference principles and practiceof knowledge discovery in the databases
9. Tanvi Sethi, Gagandeep, “An Improved Approach for the Software Defect Predict-ion using Artificial Neural Networks” Department of Computer Science,Punjabi University, Patiala, India
10. Du Zhang, “Applying Machine Learning Algorithm In Software Develop-ment,”Department of Computer Science California State University”.
11. Carlos Lucena, Nathalia Nascimento, Paulo Alencar and Donald Cowan, ' Software Engineers vs. Machine Learning Algorithms: An Empirical Study Assessing Performance and Reuse Tasks ' Cornell University,(arXiv:1802.01096v2 [cs. SE] 7 Feb 2018)
12. Kamrun Nahar Neela, Syed Ali Asif, Amit Seal Ami, Alim Ul Gias. "Modeling Software Defects as Anomalies: A Case Study on Promise Repository", Journal of Software, 2017
13. Juho Sarkamo, “Deep Learning Meth-ods for synthesis of tabular data”FEBRUARY 11 2019
14. Bartlomiej Wójcickiú, Robert Dnbrowskiú úInstitute of Informatics, University of Warsaw”Applying Machine Learning to Software Fault Predic-tion”e-Informatica Software Engineering Journal, Volume 12, Issue 1, 2018, pages: 199–216
15. Du Zhang(California State University, USA), Jeffrey J P Tsai (University of Illinois, Chicago, USA)”Series on Software Engineering and Knowledge Engineering: Volume 16”
16. Ping Yao School of Economics & Management, Heilongjiang Institute of Science and Techno-logy, Harbin, 150027, China 12308157@sina.com”Comparative study on class imbalance learning for credit scoring “2009 Ninth International Conference on Hybrid Intelligent Systems
17. Georgios Douzas, Fernando Bacao,”Effective data generation for imbalanced learning using conditional generative adversarial networks”, Expert Systems with Applications,Volume 91.
18. Anuradha Chug, Praman Deep Singh“Software Prediction Analysis Using machine Learning Algorithms”
19. David Gray, David Bowes, Neil Davey, Yi Sun and Bruce Christianson,”The Misuse of the NASA Metrics Data Program Data Sets for Automated Software Defect Prediction”
20. Rakesh Rana, Martin Nilsson “The Adoption of machine learning techniques for software fault prediction” Communication in Computer and Information Science
21. E. Fenton,Martin Neil, “A Critique of Software Defect Prediction Models”IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 25, NO. 5

AUTHORS PROFILE

S.Kaliraj , currently working as an Assistant Professor in the Department of Software Engineering at SRM Institute of Science and Technology, Chennai, India. Previously, He received a M.E degree in Software Engineering in 2013 from Anna University, Chennai, India and B.E degree in Computer Science and Engineering in 2011 from Anna University, India. And also he has completed the MCTS(Microsoft Certified Technology Specialist) Certification from Microsoft for ASP.NET Application. He has published more research papers in reputed International Journals (Scopus and SCI indexed) and Conferences. He has also organized Many National Level Conferences and International Conference. He has reviewed many research article submitted in the reputed SCI and Scopus indexed journals.His area of research includes Software Engineering, Software Testing, Application of ML in Software Engineering Disciplines. He is the Member of Indian Science Congress, Institute of Engineering and Technology.

Aman Jaiswal ,Bachelor of Technology in software engineering from SRM institute of Science and Technology, Chennai, India.He has certified courses in mind education, python and data science.Further pursuing independent research in the field of data analytics.