Improving Software Defect Prediction using Generative Adversarial Networks

P.Sampath Kumar
Asst Professor, Dept. of CSE
P.S.G College of Technology
Coimbatore, India

Dr. R.Venkatesan
Professor, Dept. of CSE
P.S.G College of Technology
Coimbatore, India

Abstract: This paper discusses Generative Adversarial networks, one of the latest techniques to solve the problem of data imbalance for software defect prediction tasks. When the data available is not enough in a task to frame a machine learning model, it is difficult to construct an accurate model. Generally Software engineering activities like defect prediction, effort estimation etc., were done on data available from open source datasets which had less data. The software defect prediction data available are not only smaller in size, but also imbalanced in nature with very less data found in the defective class. In order to overcome this data imbalance, artificial data generation techniques have been employed. In this work we try to improve the software defect prediction performance in projects, where the data available is less and imbalanced, using Generative Adversarial Networks (GANs).

Keywords: Data Imbalance, Generative Adversarial Networks, Defect Prediction, Data Augmentation, Software Engineering

1. INTRODUCTION

In Software Projects, Software Engineering (SE) tasks like effort estimation, defect prediction needs to be accurate for the projects' success. These software engineering tasks will yield better accurate results when the data available to construct the model is large enough with less class imbalance. When the product is released with large number of errors and defects, the customer will incur huge losses in terms of cost and time due to product failure and system stoppage. Due to these reasons, the need to predict the defective components at an early stage becomes crucial [1].

Machine Learning techniques are employed to predict whether the software module can be defective or not, from the previous data collected and archived. When the defective module is identified through the defective prediction process, then expert personnel can be employed to concentrate on the risky modules for early defect detection which will improve the product quality with less time and cost [2]

Information technology firms do not expose the software engineering data to outside world for conducting analysis and research. In this work, the software defect datasets from NASA Metrics Data Program (MDP) projects are used to frame the machine learning (ML) models. To get accurate results in learning models, data should be large enough and should be of good quality. Lack of data leads to generalization issues and these issues can be tackled by Data Generation techniques like GANs and Synthetic Minority Oversampling Technique (SMOTE)[3].

Data augmentation is an effective technique while dealing with a smaller dataset without over fitting. In software defect prediction projects, the data available for defect instances will be very less compared to the non-defective instances. Due to this, there is less change in accuracy, even when all the defect instances are misclassified as non-defective. In order to counter this, more data is generated for defective class using GANs and then try to improve the classification measures by combining the generated data with the real data.

2. RELATED WORK

In the last three decades, plenty of effort in research has been invested to improve the prediction of defective modules in software products. This defect prediction need to be highly accurate in order to reduce the loss incurred, due to errors occurring in the products in the customer place after release. Jones et al talks about the detecting and rectifying costs, which account for a very high amount in software project activities [4]. Researchers have used the NASA MDP Software Defect data extensively for defect classification using various machine learning algorithms. The NASA MDP data set needs to be preprocessed to yield accurate performance measures. The data needs to be cleansed and the relevant variables need to be selected to improve the performance accuracy [5].

Generative adversarial networks consist of the generator and the discriminator. The discriminator would be fed with the combination of real and generated data and it would produce an estimate for each of these inputs. The discriminator and the generator learn from each other using the min-max game through feedback [6]. Oversampling and under sampling methods can be used to handle the data imbalance. This can be achieved in defect prediction, by under sampling the majority non-defect class or oversampling the minority defect class. Chawla [8] proposed SMOTE oversampling, which creates new examples from the minority class which are closer neighbor instances in this class [7].

The generation of additional synthetic (artificial) training data will be useful when the data set is highly imbalanced. This happens where the amount of data instances for one class is less than other and this is called as minority class. Due to this issue, the learning algorithms will have higher accuracy even when all the data instances of minority classes are misclassified. To conquer this issue, minority data is augmented through the generation of data instances. This technique increases not only the defect class instances, but also it will help to avoid over fitting [9] [10]. GANs usually are used in data augmentation for computer vision tasks, where images are generated and combined with real data. In this work the architecture of GAN is modified to suit the numerical data (SDP data) and the data augmentation is done for numerical data instances of defective class.

3. THEORETICAL BACKGROUND

Data Augmentation is a useful technique to improve prediction performance measures when the data available for model creation is less. The Data Imbalance in the software defect NASA MDP datasets will cause the data model to over fit and GAN is employed to overcome the data imbalance.

3.1 Software Defect Prediction

Software Defect prediction needs to be accurate in order to detect the defective module early which will be helpful to deliver a quality product to the customer. Machine learning models are used to predict the defective modules using data history. The software firms do not reveal the software engineering process data and so the researchers use the open source NASA MDP datasets to build models.

3.2 Generative Adversarial Networks

GANs are a type of generative model, which can be used to produce new data instances based on its training data distribution. In General, when people try to do any work, they get the feedback from others and use that feedback to improve the performance. GANs consists of two entities namely a generator and a discriminator.

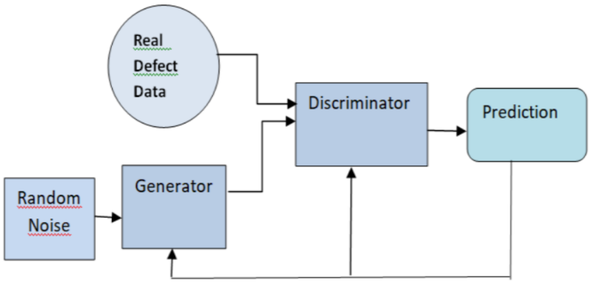

Figure 1: Generative Adversarial Networks

The generator generates the data instances from the random noise vector. The discriminator is presented with data samples from either the real data set or synthetic data generated by the generator. The discriminator makes an effort to predict the source of the data as shown in Figure 1. The generator learns to study a pattern of the data distribution it aims to generate, so that when fed with a noise vector, it predicts a sample from the estimated distribution.

In Software defect prediction (SDP), since the defect modules are found to be less than the Non-defect modules. This skewed distribution of data brings down the performance of SDP methods, where the faulty modules are not predicted accurately by the model. In this work Generative Adversarial Networks (GAN) are used to generate the additional training data for classification of software defects. This new data generated will be effective in conditions where one of the classes (defect classes) is less represented.

4. PROPOSED WORK

In this section, an effective methodology is proposed to improve the software defect prediction accuracy by using techniques which will overcome the data imbalance issue. For this work we have used the NASA MDP public Datasets with GAN techniques to illustrate how these techniques overcome the data imbalance by increasing the instances in the minority defect classes.

4.1 Data Preprocessing

In the NASA MDP Dataset there are few irregularities which need to be cleaned up before using that real data to generate the artificial data [11]. Some NASA MDP dataset contains null values and irrelevant features to be taken care. The null values are replaced by the mean values of the columns and the feature set was reduced by removing the irrelevant features and retaining only the highly relevant ones. This was done by attaching a importance score to each feature and then the top ten features were selected, based on the importance score.[14] [15].

4.2 Data Generation using GANs

Generative adversarial networks are generative models that can generate new data instances which are similar to real data. GANs have two components, called as a generator and a discriminator which plays min-max game against each other as the model is trained. The generator network tries to generate synthetic data points and discriminator tries to identify if the data generated is real one or artificially generated, as shown in Figure 2.

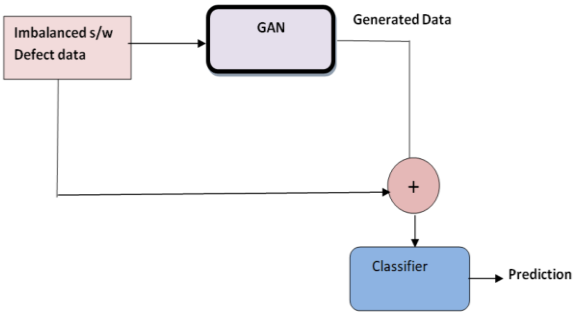

Figure 2: Flow Diagram for Data Augmentation using GANs

In GAN, the generator generates the data from the random noise. The discriminator learns from the data fed into it and from this learning, it generates an estimate of the probability that a given data was real one or not. The discriminator network would be fed with a set of data that consisted of both real and generated data and it would generate an estimate for each of these inputs. Cross-Entropy loss is used to measure the error between the actual output and the discriminator. actual output [12].

Table 1: Evaluation Metrics for NASA MDP datasets with and without GANs

Dataset	**Minority Data Generated**	**Accuracy**		**Precision**		**Recall**		**F1_Score**	
		With GANs	**With GANs**	**Without GANS**	**Without GANS**	**Without GANS**	**Without GANS**	**Without GANS**	**Without GANS**
CM1	48	92	90.6	88	86	91.6	90.6	88.13	86
PC1	146	91.5	91	85	84	92.5	91	89	88
KC1	326	91.2	91	85.8	85	92.8	91.2	88.1	88.2

GANs are used basically to generate images in Computer vision tasks using Convolutional Neural networks (CNN). Convolutional operations are utilized in the layers of the networks to understand the structure (spatial) of image data. The software defect dataset employed here is numerical in nature and does not maintain spatial structure between the features. Due to this, convolutional neural networks have been converted into fully (densely) connected neural networks [18].

5. RESULTS AND DISCUSSION

This section reports the results of the experiments conducted with the NASA MDP data using GANs. GANs are used to overcome the data imbalance issue and results of the experiments with these techniques are compared with processes without GANs and try to understand how much improvement they bring in and where they can be applied effectively.

In Software defect identification problems, the data size is usually small for the minority classes as shown in the second column of Table 1. This usually will affect the classification metrics of the final prediction. Due to that issue, GAN is employed to generate the data for the minority classes and improve the predicted performance.

In this section NASA, MDP jm1, cm1and kc1 datasets have been used to conduct experiments with GANs and without GANs. The classifications metrics have been collected from the experiments conducted and the recordings have been tabulated for the evaluation metrics using GANs in Table 1.

In this experiment the issue of data imbalance is dealt using GANs and it is observed there are improvements in classification metrics. From Table 1 it can be inferred that there is slight improvement in accuracy measure (0.8%) and a relatively better improvement in precision (2.2%), recall (2.3%) and F1_score (3.2%). Using GANs the minority class data size has been increased and found that this generated data improves the overall classification metrics.

6. CONCLUSION

In this work, the issue of data imbalance in software defect datasets (NASA MDP) is dealt by application of Generative Adversarial Networks. The usage of these techniques have allowed to overcome the data imbalance issues and marginal improvements have been observed for the classification metrics from the results obtained. The marginal improvement of (0.8-3.2)% in classification metrics is noticed with the application of plain GANs. In future, other GAN models like WCGAN, DRAGAN etc., can be applied for these data imbalance issues which have the ability to improve the classification metrics to s higher level. This technique can be applied to larger datasets collected from the software industries, which would provide us the opportunity to work on the data obtained from recently concluded projects. This will lead to a higher software defect prediction accuracy which will result in a better product delivery to the customer.

7. ACKNOWLEDGMENTS

The authors would like to thank the principal and management of PSG College of technology for their support and help rendered to complete the experiments successfully.

8. REFERENCES

[1] Y. Kamei and E. Shihab, “Defect Prediction: Accomplishments and Future Challenges,” no. 1, pp. 33–45, 2016.

[2] R. Li, L. Zhou, S. Zhang, H. Liu, X. Huang, and Z. Sun, “Software Defect Prediction Based on Ensemble Learning,” *ACM Int. Conf. Proceeding Ser.*, pp. 1–6, 2019.

[3] Wohono R.S., “Systematic Literature Review of Software Defect Prediction: Research Trends, Datasets, Methods and Frameworks,” *J. Software Engineering* vol. 1, no. 1, pp. 1–16, 2015.

[4] Jones, C and Bonsignour, O.” The Economics of Software Quality. Pearson Education, Inc.”2012.

[5] S. Agarwal and D. Tomar, “A Feature Selection Based Model for Software Defect Prediction,” *Int. J. Adv. Sci. Technol.*, vol. 65, pp. 39–58, 2014.

[6] https://www.toptal.com/machine-learning/generative-adversarial-networks.

[7] S. Zheng, A. Farahat, and C. Gupta, "Generative Adversarial Networks for Failure Prediction," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11908 LNAI, pp. 621–637, 2020.

[8] Chawla,V., Bowyer, K., Hall, L.O., Kegelmeyer, W.P.:" Smote: synthetic minority over-sampling technique."Journal of artificial intelligence research 16, 321– 357 (2002)

[9] F. H. K. dos S. Tanaka and C. Aranha, "Data Augmentation Using GANs," vol. 2019, pp. 1–16, 2019.

[10] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, "Understanding Data Augmentation for Classification: When to Warp?," *2016 Int. Conf. Digit. Image Comput. Tech. Appl. DICTA 2016*, 2016.

[11] Shepperd M, Song Q, Sun Z, and Mair C, "Data quality: Some comments on the NASA software defect datasets," *IEEE Trans. Softw. Eng.*, vol. 39, no. 9, pp. 1208–1215, 2013.

[12] Q. Liu, G. Ma, and C. Cheng, "Data Fusion Generative Adversarial Network for Multi-Class Imbalanced Fault Diagnosis of Rotating Machinery," *IEEE Access*, vol. 8, pp. 70111–70124, 2020.

[13] S. S. Mullick, S. Datta, and S. Das, "Generative adversarial minority oversampling," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-October, pp. 1695–1704, 2019.

[14] T. Iliou, C.-N. Anagnostopoulos, M. Nerantzaki, and G. Anastassopoulos, "A Novel Machine Learning Data Preprocessing Method for Enhancing Classification Algorithms Performance," pp. 1–5, 2015.

[15] S. García, J. Luengo, and F. Herrera, "Tutorial on practical tips of the most influential data preprocessing algorithms in data mining," *Knowledge-Based Syst.*, vol. 98, pp. 1–29, 2016

[16] A. Sonali and D. Siddhant, "Prediction of Software Defects using Twin Support Vector Machine", 2nd International conference on Information Systems & computer Networks (ISCON-2014), In press

[17] Fenton N.E and Pfleeger S.L, "Software metrics: a rigorous and practical approach", PWS Publishing Co., (1998).

[18] https://www.toptal.com/machine-learning/generative-adversarial-networks