Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 11, November-2019

A Novel Approach to Handle Class Imbalance in Machine Learning

Ms. Monica. Ochani
Department of Computer Engineering
Datta Meghe College Of Engineering,
Airoli, Navi Mumbai, INDIA

Dr. S. D. Sawarkar
Department of Computer Engineering
Datta Meghe College Of Engineering,
Airoli, Navi Mumbai, INDIA

Mrs. Swati Narwane
Department of Computer Engineering
Datta Meghe College Of Engineering,
Airoli, Navi Mumbai, INDIA

***Abstract*—Machine learning is the study of algorithms that a system uses to effectively perform a specific task. It depends on the patterns and inference instead of any instructions. In machine learning, majorly there is always some level of class imbalance issue in real-world classification. This problem arises when each class does not make up an equal division of a data-set. It is important to properly change the metrics and methods to balance the data set goals. This means that many learning algorithms of machine learning have low predictive accuracy for the not often occurring class. In this paper, we shall discuss this problem and look into different approaches used to solve the class imbalanced issue. This paper discusses the survey of different approaches done to improve the class imbalance issue in the data sets by learning about the data level approaches and the algorithm approaches. We have discussed the oversampling and under sampling methods to overcome the data imbalance problem.**

Keywords*—Class imbalance, data mining, machine learning, imbalance data, applications, classification, approach, algorithm, sampling.

I. INTRODUCTION

In recent years the studies have grown emphasis on class imbalance. The classification for class There are many industries which are affected by class imbalance distribution Reported works in classifications for class imbalance distribution come in many ranges of domain applications like diagnosis of faults, abnormality detection, medical diagnosis, oil spill detection in images taken by satellites, face recognition, text classification, and many others. The most important challenges of the class imbalance issue is of pattern recognition and data mining. Same problem is seen in practical applications. [3]

In machine learning, data mining is one of the most important branch. As the real world is getting exposed with new technologies the data is also increasing with increase in number of problems. These problems can be marked as volume and velocity of data. In imbalanced classes accuracy is not always true which is very standard problem classification in machine learning. There is always a difference in datasets with asymmetric ratio of observations in a class. Few examples of applications

which have imbalanced data sets are: reports of medical diagnosis, finance industry etc,. The datasets faces imbalanced class distribution when one of the classes is not sufficiently represented. Basically it means that the number of examples of the class which are less than half of the whole dataset is considerably smaller than the number of examples of the class which is sufficiently represented. [1]

If an individual has learned about machine learning and data science, he/she definitely understands that imbalanced class distribution is always seen in machine learning. This situation occurs only when the observations listed in a class is very low than that present in the other classes. This problem is primarily in applications where detection of inconsistency is crucial like electricity thievery, unauthorised transactions in banks, identification of uncommon diseases, etc. In this scenario, the predictive model developed using machine learning algorithms could be incorrect. This is mainly due to Machine Learning algorithms which are mostly made to improve correctness by minimising the error. Hence, they do not take into account the class distribution or balance of classes. [15]

II. METHODOLOGY

In this section we shall learn about the various data level approaches which are used to balance the data and eliminate the class imbalance issue. We shall also explain our proposed methodology to solve the class imbalance problem. In section A we shall explain various approaches and in section B we shall explain our proposed system to address the problem

A. Approaches to handle class imbalance

Data-level approaches are an external methods which uses the pre-processing step to rebalance the class distribution. These methods are developed and can be used at the pre-processing level. The most commonly used approach is resampling. This approach has two processes namely undersampling and oversampling of the dataset. If a dataset is balanced by removing the instances from majority class is known as undersampling. When we add similar instances of minority class to balance the class ratio then oversampling is achieved. We can do resampling with or without the replacement. We shall see each approach in details below. [1]

- **Resampling:** The process of recreating the sample data from the actual data sets is called resampling. It can be either done by non-statistical estimation or by statistical estimation. In non-statistical estimation, we randomly draw samples from the actual dataset hoping that the data is divided in a similar division to the actual dataset. However, in statistical estimation, we estimate the parameters of the actual dataset and then drawing the subsamples. Hence, we can extract data samples that carry most of the information from the actual population. Thus the resampling technique statical as well as non statistical help us in drawing the samples when the data is imbalanced.[16]
- **Undersampling:** It is a technique in which we randomly select samples from the majority class and

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 11, November-2019

discard the remaining. We assume that any random sample approximately reflect the division of the data. In this method the goal is to balance the distributions in the class through a random elimination of majority class observations. The k-nearest neighbor based approach is one of the frequent used approaches. In these approaches the sample set is selected and then is searched exhaustively in the entire dataset and it will select the k-NN and discard the other data. It is assumed that k-NN carries all the information that we need regarding those classes in this method. [16]

Many other undersampling techniques are also available which are based on two different types of noise model hypotheses. In this technique we assume that the samples near the boundary are noise. Hence the noise will be discarded in order to obtain the maximum accuracy. For another noise model, the assumption is that if the location of the majority class samples and the minority class samples are same then they are noise. If we discard these samples from the data then it creates a clear boundary that can help in classification.

- **Oversampling**: In oversampling we do this by replicating the minority samples so that the distribution is equal and balanced. One more very common approach used in oversampling is SMOTE. This method helps to overcome the shortcomings of oversampling. It creates the new samples by introducing based on the distances between the point and its nearest neighbors. SMOTE also calculates the distances for the minority samples which are near the decision boundary and generates new samples. This will affect the decision boundary to move away from the majority classes and remove the overfitting issue. [7]
  - **Random Oversampling:** Random Oversampling involves supplementing the training data with multiple copies of some of the minority classes. Oversampling can be done more than one time and is also proven to be robust. [3] Instead of duplicating every sample in the minority class, some of them may be randomly chosen with replacement.
- **SMOTE:** There are many methods available to oversample a dataset used in a typical classification problem . The most common technique is known as SMOTE - Synthetic Minority Over-sampling Technique. We consider some training data which has s samples, and f features in the feature space of the data. Note that these features, for simplicity, are continuous. Now to oversample, we shall pull a sample from the dataset and consider its k nearest neighbors. To produce a synthetic data point, consider the vector between one of the k neighbors, and the present data point. Now multiply the vector by a random number that lies between 0, and 1. Adding this to the present datapoint new synthetic data point will be created. This way SMOTE can be modified extended to eliminate the imbalance dataset.[7]

- **ADASYN:** The full form of it is Adaptive Synthetic Sampling Approach. It expands on the procedure of SMOTE, by shifting the importance of the classification boundary to those minority classes which are difficult. ADASYN uses a weighted spread for not similar minority class examples as per their level of complexity in learning, where more synthetic data is created for minority class examples that are difficult to learn.

B. Proposed System

Many systems and approaches have been proposed which handle the imbalance dataset. However we still have issues with the dataset and class imbalance , hence we propose

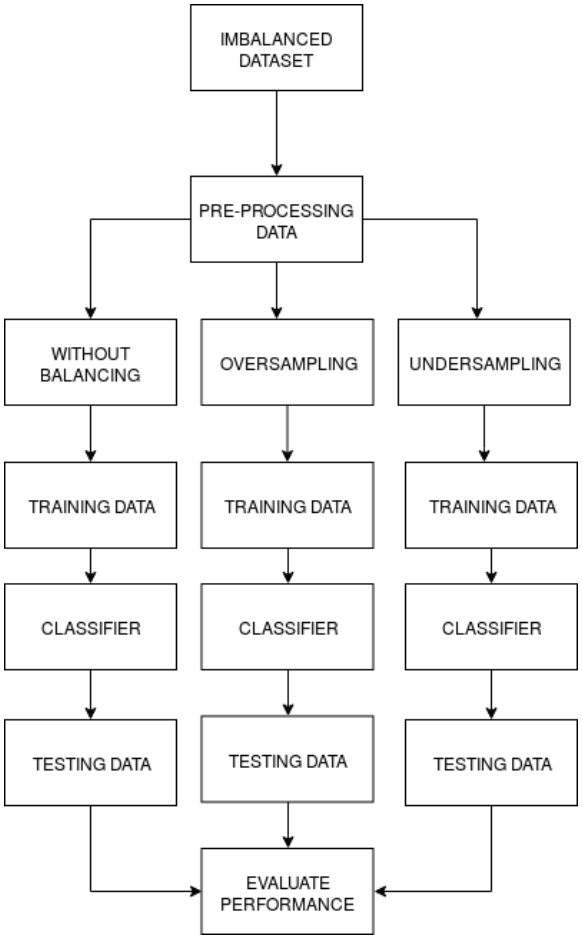

Fig 1. Proposed System Block diagram

There are a number of methods available to oversample a dataset used in a typical classification problem (using a classification algorithm to classify a set of images, given a labeled training set of images). The most common technique is known as SMOTE. We have already explained it in the above section.

NearMiss is an under-sampling technique. In this method instead of resampling the minority class, using a distance, it will make the majority class equal to the minority class. There are three near-miss methods: The first method NearMiss-1 where the majority class samples are selected while their average distances to three closest minority class samples are the smallest. The second method NearMiss-2 the majority class samples are selected while their average distances to three farthest minority class samples are the smallest. The third method NearMiss-3 gives number of the closest majority class samples for each minority class sample. [21]

A training dataset is a sample dataset used for understanding, so it fits the parameters (e.g., weights)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 11, November-2019

of, for sample, a classifier. Most approaches that search through training data can recognize and utilize apparent relationships in the training data that do not hold in general. A test dataset is a dataset that is free of the training dataset, but that follows the same probability spread as the training dataset.

There are different types of classifiers, a classifier is an algorithm that maps the input data to a specific category. We shall list out the classifiers, we shall use any one of the following list for our system to balance out the data. We have listed out the types of classification algorithms in Machine Learning:

- Linear Classifiers: Logistic Regression, Naive Bayes Classifier
- Support Vector Machines
- Decision Trees
- Boosted Trees
- Random Forest
- Neural Networks
- Nearest Neighbor

We shall take data to follow all the steps which are explained in the block diagram and then compare the data which is not sampled and see data imbalance in that data. After which we shall use the data-level approaches of oversampling and undersampling compare the results of each of them to solve the class imbalance problem. When the data is not sampled, we should experience the class imbalance problem. After using the oversampling and undersampling approaches for the same data we shall see the elimination of class imbalance. For model evaluation, we shall use AUC, based on which we can decide whether our model is performing better after balancing the data.

Area Under Curve(AUC) is one of the most widely used metrics for evaluation. It is used for binary classification problem. AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example.

- True Positive Rate (Sensitivity) : True Positive Rate is defined as - TP / (FN+TP).
- True Positive Rate corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points.
- True Positive Rate = True Positive / False Negative + True Positive
- False Positive Rate is defined as - FP / (FP+TN).
- False Positive Rate corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points.
- False Positive Rate = False Positive / False Positive + True Negative

False Positive Rate and True Positive Rate both have values in the range [0, 1]. FPR and TPR both

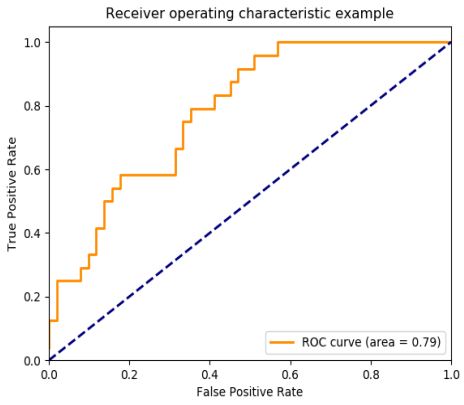

Fig 2. Receiver Operating characteristic example [21]

are computed at threshold values such as (0.00, 0.02, 0.04, …., 1.00) and a graph is drawn. AUC is the area under the curve of plot False Positive Rate vs True Positive Rate at different points in [0, 1]. As evident, AUC has a range of [0, 1]. The greater the value, the better is the performance of our model.[21]

III. SYSTEM OVERVIEW

System design is used for understanding the construction of system. We have explained the flow of our system and the software used in the system in this chapter.

A. Flow of the system

The **Fig. 3** explains the flow chart of the system design, we shall explain each of the components of the flow chart in each section below. As already seen in the previous sections SMOTE will generate synthetic examples instead of applying a simple duplication of the minority class instances. This new data is generated by interpolation between several minority class instances that are within a said neighborhood. The minority class is now over-sampled by taking every minority class sample and inserting synthetic examples along with the line segments joining any or all the k minority class nearest neighbors. This technique effectively forces the decision region of the minority class to become more general. [21]

It's a better version of SMOTE. It is similar to SMOTE only with a minor improvement. After generating the synthetic sample it adds random small values to the points and makes it more realistic. Hence, the samples have a little more variance in them i.e they are a bit scattered. ADASYN finds the k-nearest neighbors for each of the minority observations and computes an r value:

$$r_i = \frac{\#majority}{k}$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 11, November-2019

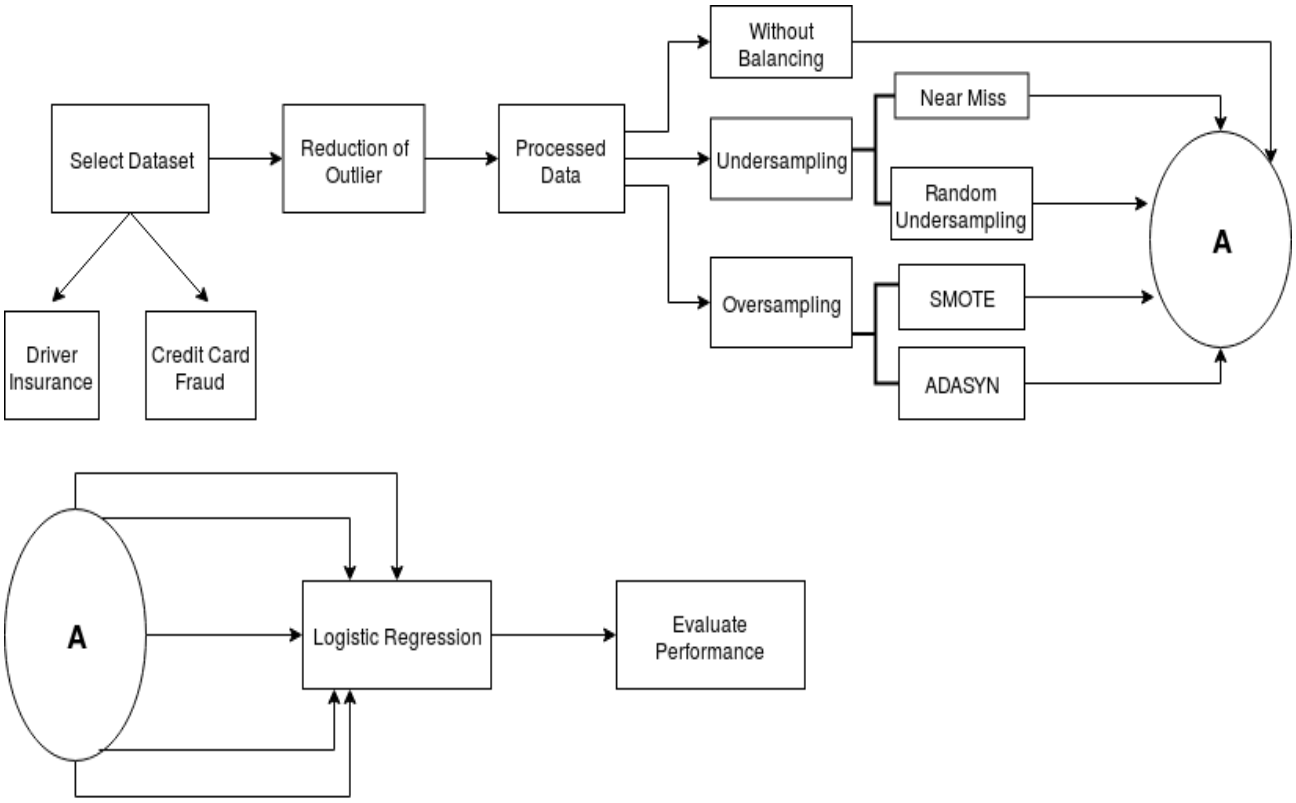

Fig 3: Flow diagram of system

The r_i value measures the dominance of the majority class in the neighborhood. The higher r_i, the more dominant the majority class and the more difficult the neighborhood is to learn for your classifier.

As discussed in the earlier sections NearMiss is an under-sampling approach. It focuses to balance class distribution by the elimination of the majority class examples randomly. When samples of two different classes are very near to each other, we remove the samples of the majority class which increases the spaces between the two classes. This aid is the classification process. Now to prevent the problem of information loss we use, near-neighbor methods which is a under sampling technique

Random under sampling removes samples randomly with or without replacement from the majority class. This is one of the preliminary methods used to eliminate the imbalance in the dataset, however, it may eliminate useful or important samples as it increases in the variance of the classifier

Logistic Regression is mostly the first machine learning algorithm that every data scientist knows. The aim of a logistic regression model is to find a relationship between one or more features which are the independent variables and a continuous target variable which are the dependent variable.

IV. IMPLEMENTATION

This section provides knowledge about the implementation environment and throws light on the actual steps for the implementation of dataset to test the eliminate the class imbalance issue.

A. Hardware requirements

The following hardware was used for the implementationof the system:

- 4 GB RAM
- 10GB HDD
- Intel 1.66 GHz Processor Pentium 4

B. Software requirements

The following software was used for the implementationof the system:

- Windows 7
- Python 3.6.0
- Visual Studio Code

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 11, November-2019

C. Implementation steps

We shall first check the level of imbalance present in our original dataset. Most of the transactions are non-fraud. If we use this data frame as the base for our predictive models and analysis we might get a lot of errors and our algorithms will probably overfit since it will "assume" that most transactions are not a fraud.

1. Here, we have taken 2 imbalanced datasets i.e. Credit card fraud & Driver Insurance.
2. We have trained four classifiers & found Logistic Regression works better in our case than other three. Hence,we have used Logistic Regression as a baseline classifier.
3. Two undersampling techniques i.e. Nearmiss, Random undersampling and two oversampling techniques i.e. SMOTE & ADASYN have been used for balancing the datasets.
4. Using Logistic Regression classifier & above balancing techniques we found average precision-recall score as a metric.
5. Its been observed that Oversampling works better than Undersampling
6. Average precision-recall score in case of Credit card fraud is better than in case of Driver Insurance.
7. Please see below Figures to understand the flow of the implementation.

We select a dataset as explained earlier in point number 1. We can see in the figure that we have selected Credit Card Fraud case.

Fig. 4: Selection of the credit card fraud dataset.

Now, we will check the dataset in terms of how imbalance our dataset is. See the graph below.

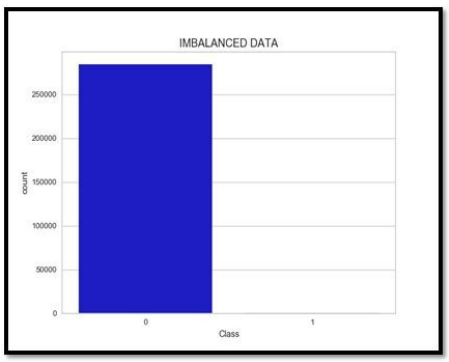

Fig 5. Graph of Imbalanced Dataset

We take four types of classifiers and chose which classifier will be more efficacious in detecting fraud transactions. From our study, we have found that the Logistic Regression classifier is more precise than the other three classifiers. Then we plot the ROC-AUC curve for all the classifiers. We then use these 2 classifiers i.e. logistic regression and random forest classifier with all the sampling techniques to get good results.

See **Fig 6**. for the ROC-AUC curve for Logistic Regression

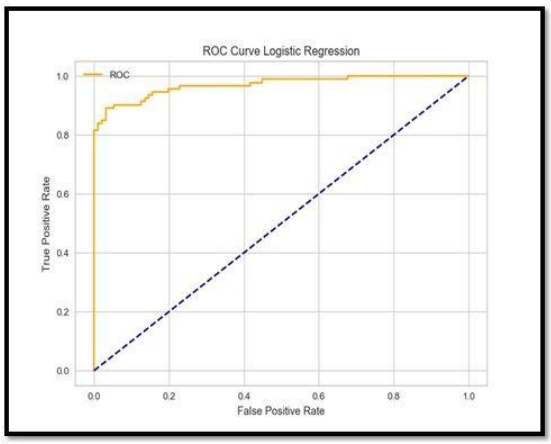

Fig 6. ROC-AUC for LR

We will now apply the oversampling technique to balance our data. We use ADASYN and SMOTE approaches under oversampling to get our final output. The figure **Fig. 7** shows graph of balanced data using SMOTE.

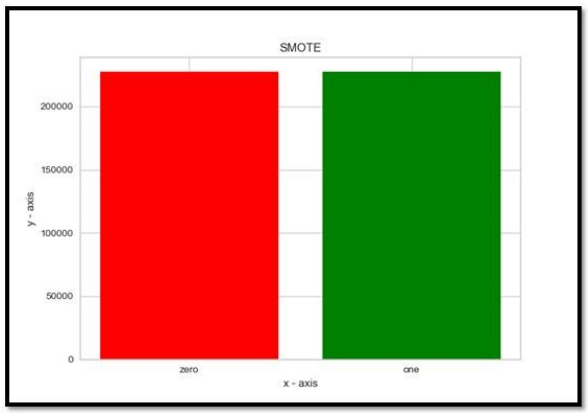

Fig 7: SMOTE: Balanced data

We get the average precision recall for SMOTE. Please see Fig 8 below.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 11, November-2019

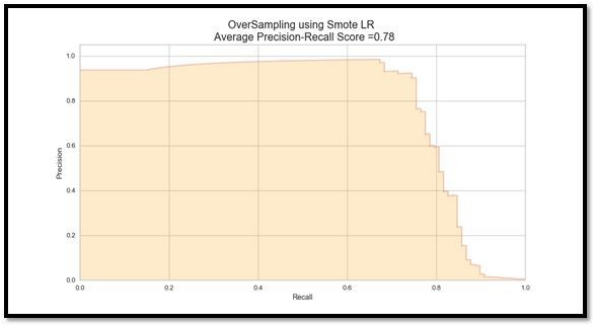

Fig 8: Average precision recall: SMOTE

We have followed the similar steps for Random undersampling, using NearMiss and Oversampling technique for ADASYN. We can see Fig 9. which explains the avaerage precision recall of all the approches.

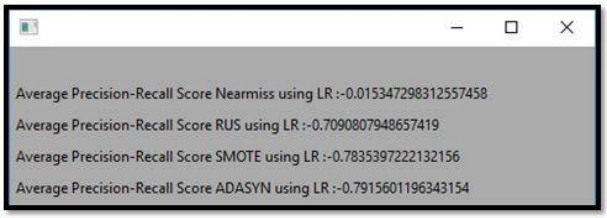

Fig 9. Average Precision for all the techniques

V. OBSERVATIONS AND RESULTS

A. *Observation*

As discussed in the earlier sections, we have used two dataset and have implemented techniques of undersampling and oversampling to get the balanced data. Below are the observations. Comparisons of different balancing techniques with the Logistic Regression as the base line classifier has been done. We have shown the average precision recall score for each.

Table 1: Observation table

Data Sets	Classifier	Balancing Techniques			
		Near Miss	Random Undersampling	SMOTE	ADASYN
Credit Card Fraud	Logistic Regression	0.015	0.709	0.783	0.791
Driver Insurance		0.050	0.058	0.056	0.056

B. *Results*

We have got the desired results of balanced dataset from an imbalanced dataset after applying different balancing technique. Refer the graph in **Fig.10** for the results. In the graph, we have shown the x-axis as the techniques or the methods used to balance the data-set and y-axis gives us information about the average precision recall score. The two datasets namely credit card fraud - highlighted in blue color and driver insurance - highlighted in red color are shown in the graph.

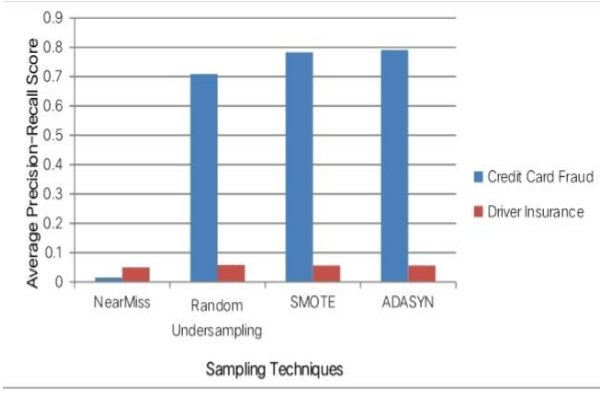

Fig 10. Results

VI. CONCLUSION AND FUTURE SCOPE

A. *Conclusion*

To conclude, we have discussed the class imbalance problem and look into different approaches used to solve it. We have also explored many different methods and algorithms to improve the class imbalance in the data sets, this includes learning about the data level approaches and the algorithm approaches. In the proposed system, we have proposed the method of oversampling and undersampling methods can be used for tackling the imbalance class problems. We have learned about the oversampling and undersampling techniques, we came to know that only after using any of the above-mentioned methods, we can overcome the problem of data imbalance. We find the novel approach to address the issue of class imbalance is sampling, oversampling and undersampling can be used to take care of class imbalance problem.

B. *Future scope*

We can take several possible directions in the future.We would like to extend our study to multi-class problems.All the methods proposed in this paper addresses two-class cases so far.Even, we can use Ensemble learning on a large collection of real datasets.In addition, we can extend our work in Big Data domain.Here, we have worked on data level approaches & would plan to conduct experimental evaluation on algorithm level approaches.

REFERENCES

[1] Ali, A., Shamsuddin, S. M., & Ralescu, A. L. (2015). Classification with class imbalance problem: a review. Int. J. Advance Soft Compu. Appl, 7(3), 176-204.

[2] Bennin, K. E., Keung, J., Phannachitta, P., Monden, A., & Mensah, S. (2017). Mahakil: Diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction. IEEE Transactions on Software Engineering, 44(6), 534-550.

[3] Gong, L., Jiang, S., Bo, L., Jiang, L., & Qian, J. (2019). A Novel Class-Imbalance Learning Approach for Both Within-Project and Cross-Project Defect Prediction. IEEE Transactions on Reliability.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 11, November-2019

[4] Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., & Seliya, N. (2018). A survey on addressing high-class imbalance in big data. Journal of Big Data, 5(1), 42.
[5] Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1249.
[6] Luo, C. (2018). A comparison analysis for credit scoring using bagging ensembles. Expert Systems, e12297.
[7] Zhu, B., Baesens, B., Backiel, A., & Vanden Broucke, S. K. (2018). Benchmarking sampling techniques for imbalance learning in churn prediction. Journal of the Operational Research Society, 69(1), 49-65.
[8] Japkowicz, N., & Stephen, S. (2016). The class imbalance problem: A systematic study. Intelligent data analysis, 6(5), 429-449.
[9] Hualong Yu, Changyin Sun Qi Wang, Xiaoyan Xi.(2018). A Fast and Flexible Cost-Sensitive Learning Framework for Classifying Imbalanced Data,IEEE
[10] Neelam Rout.(2018) Handling Imbalanced Data: A Survey,Research Gate
[11] Wei, W., Li, J., Cao, L., Ou, Y., & Chen, J. (2013). Effective detection of sophisticated online banking fraud on extremely imbalanced data. World Wide Web, 16(4), 449-475.
[12] Khor, K. C., Ting, C. Y., & Phon-Amnuaisuk, S. (2014). The effectiveness of sampling methods for the imbalanced network intrusion detection data set. In Recent Advances on Soft Computing and Data Mining (pp. 613-622). Springer, Cham.
[13] Seo, J. H., & Kim, Y. H. (2018). Machine-Learning Approach to Optimize SMOTE Ratio in Class Imbalance Dataset for Intrusion Detection. Computational Intelligence and Neuroscience, 2018.
[14] Picek, S., Heuser, A., Jovic, A., Bhasin, S., & Regazzoni, F. (2018). The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations.
[15] Cordón, I., García, S., Fernández, A., & Herrera, F. (2018). Imbalance: oversampling algorithms for imbalanced classification in R. Knowledge-Based Systems, 161, 329-341.
[16] Tsai, C. F., Lin, W. C., Hu, Y. H., & Yao, G. T. (2019). Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. Information Sciences, 477, 47-54.
[17] Santiso, S., Casillas, A., & Pérez, A. (2018). The class imbalance problem detecting adverse drug reactions in electronic health records. Health informatics journal, 1460458218799470.
[18] Agrawal, K., Baweja, Y., Dwivedi, D., Saha, R., Prasad, P., Agrawal, S., ... & Dutt, V. (2017, December). A Comparison of Class Imbalance Techniques for Real-World Landslide Predictions. In 2017 International Conference on Machine Learning and Data Science (MLDS) (pp. 1-8). IEEE.
[19] Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks, 106, 249-259.
[20] Wang, S., Minku, L. L., & Yao, X. (2018). A systematic study of online class imbalance learning with concept drift. IEEE transactions on neural networks and learning systems, 29(10), 4802-4821.
[21] https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/
[22] https://www.datascience.com/blog/imbalanced-data
[23] https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861
[24] https://elitedatascience.com/imbalanced-classes
[25] https://www.analyticsvidhya.com/blog/2016/02/7-important-model-evaluation-error-metrics/
[26] https://towardsdatascience.com/detecting-financial-fraud-using-machine-learning-three-ways-of-winning-the-war-against-imbalanced-a03f8815cce9
[27] https://onlinelibrary.wiley.com/doi/pdf/10.1002/sec.564
[28] https://sci2s.ugr.es/keel/pdf/specific/congreso/yen2006a.pdf
[29] https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-106
[30] https://numpy.org/neps7
[31] https://numpy.org/doc/