Performance Evaluation of Class Balancing Techniques for Credit Card Fraud Detection

Dilip Singh Sisodia, Nerella Keerthana Reddy, Shivangi Bhandari
Department of Computer Science & Engineering
National Institute of Technology Raipur
Raipur, India

Abstract— **The number of online transactions has unraveled in large proportions with each passing day. Credit card transactions constitute a huge portion of these transactions. The financial losses have also increased analogously along with the credit card fraud transactions. Therefore, fraud detection systems have acquired great importance for banks and financial institutions. As the occurrence of fraud is unlikely in comparison to normally occurring transactions, we are posed with the class imbalance problem and to handle this imbalance problem we use resampling techniques in this paper. We applied oversampling (SMOTE, SMOTE ENN, SAFE SMOTE, ROS, SMOTE TL). On the resampled data, we applied cost sensitive (CSVM, C4.5) and ensemble classifier (Adaboost, Bagging) to evaluate the performances using sensitivity, specificity, G-mean, Area under ROC. We observed that the SMOTE ENN method detects the fraud in a better way than other classifiers in the set of oversampling techniques considered, and TL works better on the set of undersampling techniques taken.**

Keywords—class imbalance, cost sensitive, classifiers, ensemble learners, fraud detection, sampling, SMOTE.

I. INTRODUCTION

Fraud has been increasing drastically with the progression of state-of-art technology and worldwide communication[1]. Fraud can be curbed in two ways: prevention and detection. Prevention of data is where a layer of protection is formed to avoid any attacks from the outsiders. It tries to stop fraud from occurring in the first place. In contrast, fraud detection helps in identifying and alerting as soon as it has been perpetrated. So, detection comes into the scene once the prevention has already failed. So, detection must always be running as no one can predict when a breach might occur to the protection given by fraud prevention techniques[2], [3].

Credit card fraud is a critical problem and has accountable importance for financial companies. So, it is imperative to improve fraud detection methods along with security modules that try to prevent fraud. Fraud detection systems are trained using older transactions to decide about future ones. The faster a fraud detection system performs, the better.

In fraud detection, the count of normal cases is much more than the unauthorized cases. This causes a condition called “imbalanced data” where one class of data has a very high number of instances as compared to the other class of data. This leads us to the “class imbalance problem”. The most standard machine learning techniques assume or expect the balanced distribution of class[4]. For solving the problem of learning from data sets which are imbalanced, many solutions have been stated in the past few years. Most popular proposed solutions roughly fall into three groups: data-level, algorithm-level, and ensemble solutions. Data-level solutions apply resampling as a preprocessing step to reduce the negative effect caused by class imbalance. Algorithm-level solutions aim to develop new algorithms or modify existing ones to bias learning towards the minority class. Ensemble solutions either modify the ensemble learning algorithms at the data-level to preprocess the data before the learning stage of base classifiers or add a cost-sensitive framework in the process of ensemble learning [5].

In this paper, two types of resampling methods are used oversampling and undersampling for balancing the majority and minority classes.

The remainder of this paper is structured as follows: Section 2 presents the literature review consisting of background and related work. Section3 is about the methodologies containing process flow along with resampling and data mining techniques. Section 4 describes the experimental setup and the results. Lastly, section 5 concludes this paper.

II. RELATED WORK

A. Background

Accurate assessments cannot be made as only checkout data is looked at to make fraud assessment limiting its ability. There is a limited set of options and is susceptible to false positives. A lot of these rules are known to sophisticated fraudsters hence weakening them. The cost can curb extensive research in this area [2]. Also, it’s a relatively new area of research that has been improving alongside the boost of internet and technology. So, the research already done is low when compared to other fields. Some models cannot identify frauds that are obvious to the human eye.

The biggest issue could be the non-availability of dataset because most financial institutes are hesitant in giving up such confidential information for research. Even if the data set is available, they are imbalanced. There are also millions of transactions taking place in a day, and such huge data is difficult to process and train as mentioned in [8].

978-1-5386-0814-2/17/$31.00 ©2017 IEEE

B. *Related Work*

Credit card fraud detection has been a matter of interest as the technology kept growing. A lot of institutes and individuals have incurred losses that ranged from hundreds to millions of dollars throughout the globe. As the detection and prevention techniques increased, so did the sophistication of fraudsters. To curb such fraudsters, extensive research has started to take place.

The first issue that needed solving was the imbalanced learning problem. Imbalanced data occurs due to the rarity of occurrence of a class of data in the dataset. Even though the problem of credit card fraud is huge, the ratio of fraudsters to legitimate users is very much low. This results in imbalanced data.

There are a lot of ways to handle imbalanced data. According to [6], when standard machine learning techniques are used to the imbalanced data, the induction rules that define the majority concepts are often stronger than those of minority concept. According to [9], class-imbalance is studied in a distributed environment on a large scale sparse data. Here it is treated as a cost-sensitive learning problem. Sampling methods are also used in [10], where it is shown that the appropriate preparation of the dataset to be analyzed is the main step for making correct calculations and carrying out a precise prediction. Oversampling techniques such as MTDF, synthetic minority over-sampling technique (SMOTE), ADASYN, MWMOTE, etc. are used for balancing and preparation of large, imbalanced datasets as shown in [11]. Authors in [7] use churn prediction to tackle the imbalance problem.

There is quite some research happening in credit card fraud detection. In [2], the authors have used Artificial Immune System for better accuracy and speed in detecting the fraud detection. In [3], survey Account Signature which steps toward real-time fraud detection. According to [8], one of the best techniques is ensemble learning because of their extraordinary predictive performance on real-life problems. While constructing a credit card fraud detection model, it is essential to extract the right features and characteristics from the transactional credit card data. In [12], the authors use von Mises distribution to increase savings by proposing a new measure in which comparison of the financial cost of a technique is done with a model having none. After that, they came up with an illustrating version of the transaction aggregation strategy, by including a combination criterion when clubbing transactions. There are also a lot of works using different combinations of classifiers and methods to see the best efficiency and speed of detection system.

III. METHODOLOGY

A. *Data Description*

The credit card fraud data set is collected from Kaggle [6]. The working on whole data set at a time generating the process overhead. Therefore, the dataset is randomly divided into three small datasets and named them as DS1 DS2, DS3 consisting of ten thousand instances, fifteen thousand instances and twenty thousand instances respectively to generalize the result. To pre-process our data, we have used 5*2 cross-validation technique in which the 2-fold cross validation is performed five times.

B. *Process flow*

In this paper, the problem of class imbalance is addressed by using the different methods of sampling and some cost-sensitive classifiers. In the process flow (Fig.1), we first extract the three data sets as required. After that, we applied two types of samplings on each of them- oversampling and undersampling. In oversampling, we have used five different types, and we used four types of undersampling. Then to classify the data, we have used three types of classifiers- cost-sensitive C4.5 decision tree, AdaBoost, and bagging. To classify the output of undersampling, we used four classifiers- the three which we have used with oversampling and cost-sensitive support vector machines additionally. Then, we analyze the performance of each of the model based on four types of performance metrics and decide which model has given the best performance.

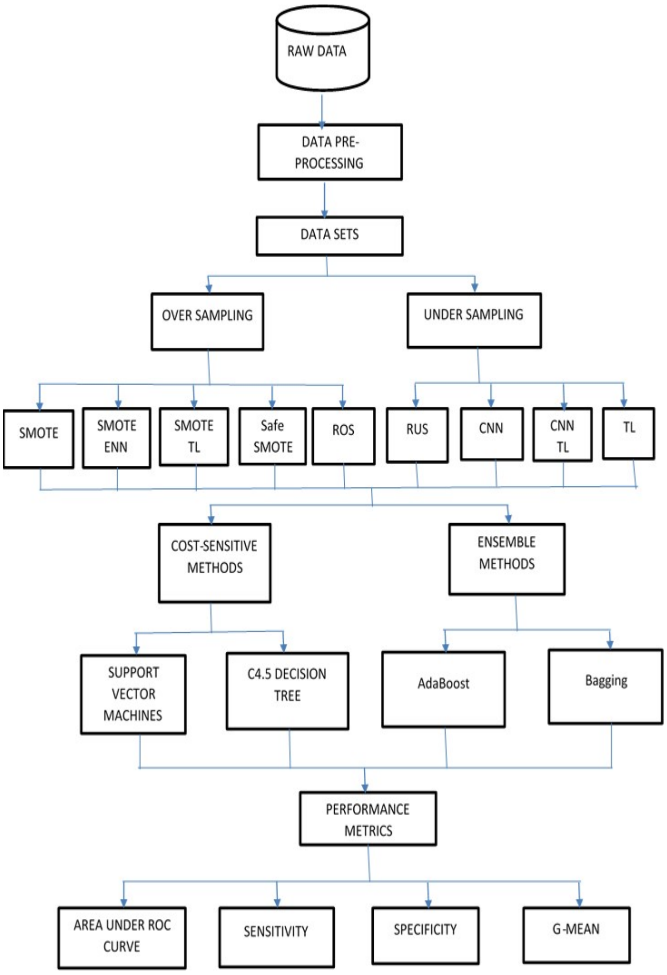

Fig. 1. The process flow of present work

C. Resampling Strategies

In this paper, we carried out oversampling by increasing the number of minority instances and undersampling by decreasing the number of majority instances.

We used SMOTE, SMOTE ENN, SMOTE TL, SAFE SMOTE, ROS among oversampling techniques. In SMOTE the minority class is over-sampled by creating "synthetic" examples rather than using replacement. In [7], the SMOTE is discussed extensively. SMOTE-ENN is Synthetic Minority Over-Sampling Technique with Edited Nearest Neighbors. In [8], the motivation behind this method is explained. We learn from[9] that Based on SMOTE, Safe-Level-SMOTE assigns each positive instance its safe level before generating synthetic instances Among undersampling methods we applied random undersampling (RUS), condensed nearest neighbor (CNN), CNN TL, TL. RUS is a non-heuristic method that aims to balance class distribution through the random elimination of majority class examples [8]. In CNN the database is summarized by finding only the important data-points [10].TL is Tomek's modification of CNN [11]

D. Data Mining Techniques

We applied cost sensitive and ensemble classifiers to the resampled data. The cost sensitive classification technique incorporates the notion of cost into the design explicitly[12]. In Cost-sensitive classification the costs caused by different kinds of errors are not assumed to be equal, i.e., they are given weights or costs, and the objective is to minimize the expected costs. C4.5 decision tree and CSVM classifiers were used in this category. A cost-sensitive decision tree approach is used for the detecting fraud in[13]. A normal SVM algorithm may do a lot of misclassifications while trying to increase the accuracy of the dataset. To overcome this, the cost-sensitive approach (csvm) is used, in which some errors are given more importance as compared to others[14].

In ensemble classification, we use modified versions of existing classifier algorithms to make them suitable for the problem of class imbalance. Adaboost and bagging were used in this category. In AdaBoost, we use an iterative process to improve the simple boosting process [15]. It focuses on the patterns that are hard to classify. In bagging [38], the training subsets are randomly drawn (with replacement) from the training set. Bagging tries at increasing accuracy by creating an improved aggregated classifier, *I**, by joining various outcomes of classifiers into a single prediction[16].

E. Performance Measures

On the resampled data, we applied cost sensitive and ensemble classifiers to evaluate the performances based on the confusion matrix as shown in Table 1.

TABLE I. CONFUSION MATRIX

		Predicted Class	
A c		FALSE	TRUE
	FALSE	True Negative (TN)	False Positive (FP)
	TRUE	False Negative (FN)	True Positive (TP)

The performance metrics which we have used in dealing with our problem are sensitivity, specificity [17], G-Mean. [18] and area under the ROC curve, which is used in [19] for credit card fraud detection. Sensitivity, sometimes called as True Positive Rate is a measure of the fraction of positively classified instances, of a model. A high value of sensitivity indicates less number of false negatives, and vice-versa. This is often at odds with specificity. That is, attempting to increase sensitivity would decrease specificity and attempt to increase specificity decreases sensitivity. ROC curve stands for Receiver operating characteristic curve, which is obtained by plotting the True positive rate against false positive rate at different settings of the threshold. The relative trade-offs between true positives and false positives are depicted by the ROC graph [20].

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$G - Mean = \sqrt{(SN \times SP)}$$

IV. EXPERIMENTS AND RESULTS

In the following Section, the experimental results that drive towards our conclusion are presented. The number of frauds, number of non-frauds, and the imbalance ratio present in our datasets is tabulated in Table 2.

TABLE II. DATASETS WITH DIFFERENT IMBALANCE RATIO

Dataset Name	**Number of instances**	**Number of Frauds**	**Number of non-frauds**	**Imbalance ratio**
DS1	10000	38	9961	0.00381488
DS2	15000	50	14950	0.00334448
DS3	20000	53	19947	0.00265704

To pre-process our data, we have used 5*2 cross-validation technique in which the 2-fold cross validation is performed five times. On this preprocessed data resampling techniques were applied followed by classification methods. The experimental results are shown in Table 3 and 4 and graph 2 to 5..

TABLE III. PERFORMANCE OF OVERSAMPLING TECHNIQUES

			AUC-ROC	**SP**	**SN**	**G-MEAN**
SMOTE	C4.5 CS	DS 1	0.84445	0.856741	0.842105	0.8493
		DS 2	0.81337	0.706755	0.92	0.8063
		DS 3	0.86596	0.81009	0.9425	0.8737
	ADA	DS 1	0.92689	0.993072	0.868421	0.9286

	BOO ST	DS 2	0.91615	0.992307	0.84	0.9129
		DS 3	0.94943	0.99704	0.90566	0.9502
	BAG GING	DS 1	0.93954	0.993374	0.894736	0.9427
		DS 2	0.92789	0.995785	0.86	0.9254
		DS 3	0.95842	0.99684	0.92452	0.96
SMOTE ENN	C4.5 CS	DS 1	0.93428	0.925710	0.947368	0.9364
		DS 2	0.91357	0.997123	0.8	0.8931
		DS 3	0.92498	0.92815	0.92452	0.9263
	ADA BOO ST	DS 1	0.95493	0.995582	0.921052	0.9575
		DS 2	0.95749	0.994983	0.92	0.9567
		DS 3	0.94943	0.99704	0.90566	0.9502
	BAG GING	DS 1	0.96600	0.989157	0.947368	0.9680
		DS 2	0.93561	0.991237	0.88	0.9339
		DS 3	0.94787	0.99393	0.90566	0.9487
SMOTE TL	C4.5 CS	DS 1	0.92185	0.929424	0.921052	0.9252
		DS 2	0.89374	0.887491	0.9	0.8937
		DS 3	0.92158	0.92134	0.924528	0.9229
	ADA BOO ST	DS 1	0.95508	0.995883	0.921052	0.9577
		DS 2	0.92745	0.994916	0.86	0.9250
		DS 3	0.95789	0.99578	0.924528	0.9594
	BAG GING	DS 1	0.96686	0.990864	0.947368	0.9688
		DS 2	0.93682	0.993645	0.9	0.9456
		DS 3	0.93867	0.99393	0.90566	0.9487
SAFE SMOTE	C4.5 CS	DS 1	0.83419	0.961248	0.710526	0.8264
		DS 2	0.82521	0.910434	0.74	0.8208
		DS 3	0.75180	0.93633	0.566037	0.728
	ADA BOO ST	DS 1	0.83130	0.984037	0.684210	0.8205
		DS 2	0.83180	0.983612	0.68	0.8178
		DS 3	0.74082	0.97438	0.5094	0.7045
	BAG GING	DS 1	0.85924	0.986346	0.736842	0.8525
		DS 2	0.85468	0.989364	0.72	0.8440
		DS 3	0.72502	0.97914	0.47169	0.6795
ROS	C4.5 CS	DS 1	0.92427	0.929424	0.921052	0.9252
		DS 2	0.89856	0.997123	0.8	0.8931
		DS 3	0.75180	0.93633	0.566037	0.728
	ADA BOO ST	DS 1	0.95623	0.998192	0.921052	0.9588
		DS 2	0.86943	0.998862	0.74	0.8597
		DS 3	0.91256	0.92134	0.924528	0.9229
	BAG GING	DS 1	0.97042	0.997992	0.947368	0.9723
		DS 2	0.87876	0.997525	0.76	0.8707
		DS 3	0.91256	0.92134	0.924528	0.9229

In Table 3, results show that SMOTE ENN performed the best among all oversampling methods. In most of the cases bagging performed better among classifiers.

TABLE IV. PERFORMANCE OF OVERSAMPLING TECHNIQUES.

			AUC-ROC	SP	SN	G-MEAN
RUS	C4.5 CS	DS 1	0.840928	0.76749	0.92105	0.8407
		DS 2	0.93953177	0.89906	0.98	0.9386
		DS 3	0.48098429	0.00015	0.9622	0.0117
	ADAB OOST	DS 1	0.94303646	0.97179	0.92105	0.9460
		DS 2	0.94408027	0.90816	0.98	0.9433
		DS 3	0.94987022	0.9597	0.94339	0.9515
	BAGGI NG	DS 1	0.93831877	0.96235	0.92105	0.9414
		DS 2	0.92685619	0.91371	0.94	0.9267
		DS 3	0.95257979	0.9851	0.9245	0.9543
	CSVM CS	DS 1	0.96575706	0.98865	0.94736	0.9677
		DS 2	0.9129097	0.94581	0.88	0.9123
		DS 3	0.95446029	0.9889	0.9245	0.9561
CNN	C4.5 CS	DS 1	0.59447791	0.18893	1	0.4346
		DS 2	0.62264214	0.34528	0.9	0.5574
		DS 3	0.48098429	0.00015	0.9622	0.0117
	ADAB OOST	DS 1	0.94745508	0.98062	0.92105	0.9503
		DS 2	0.88123746	0.96247	0.8	0.8774
		DS 3	0.90254635	0.91963	0.88679	0.9030
	BAGGI NG	DS 1	0.9156197	0.99909	0.84210	0.9172
		DS 2	0.85979933	0.97959	0.74	0.8514
		DS 3	0.95855287	0.99528	0.9245	0.9591
	CSVM CS	DS 1	0.95330714	0.98875	0.92105	0.9543
		DS 2	0.83816054	0.95632	0.72	0.8297
		DS 3	0.90379354	0.99849	0.811	0.897
CNN TL	C4.5 CS	DS 1	0.55623924	0.19817	0.92105	0.4272
		DS 2	0.7126087	0.48521	0.94	0.6753
		DS 3	0.56292303	0.01442	1	0.0120
	ADAB OOST	DS 1	0.83998385	0.76568	0.92105	0.8397
		DS 2	0.89648829	0.93297	0.86	0.8957
		DS 3	0.90235778	0.8828	0.9245	0.9034
	BAGGI NG	DS 1	0.91401327	0.97088	0.86842	0.9182
		DS 2	0.86822742	0.93645	0.8	0.8655
		DS 3	0.87353161	0.87791	0.8679	0.7619
	CSVM CS	DS 1	0.96628728	0.96114	0.97368	0.9673
		DS 2	0.82849498	0.95699	0.7	0.8184
		DS 3	0.95293911	0.98405	0.9245	0.9097
TL	C4.5 CS	DS 1	0.97002297	0.99718	0.94736	0.9719
		DS 2	0.92725753	0.99451	0.86	0.9248
		DS 3	0.48098429	0.00015	0.9622	0.0117
	ADAB OOST	DS 1	0.93020654	0.99969	0.86842	0.9317
		DS 2	0.90983278	0.99966	0.82	0.9053
		DS 3	0.9608339	0.9984	0.9245	0.9614
	BAGGI NG	DS 1	0.94265634	0.99959	0.89473	0.9457
		DS 2	0.86973244	0.99946	0.74	0.8600
		DS 3	0.95161765	0.99959	0.90566	0.9514
	CSVM CS	DS 1	0.97122777	0.99959	0.94736	0.9731
		DS 2	0.83816054	0.95632	0.72	0.8297
		DS 3	0.95446029	0.9889	0.9245	0.9561

From Table 4, it is observed that TL performed the best among all oversampling methods. In most of the cases cost, sensitive support vector machine (SVM) performed better among classifiers.

The following graphs depict the Area under ROC and G-means of different oversampling and sampling methods used.

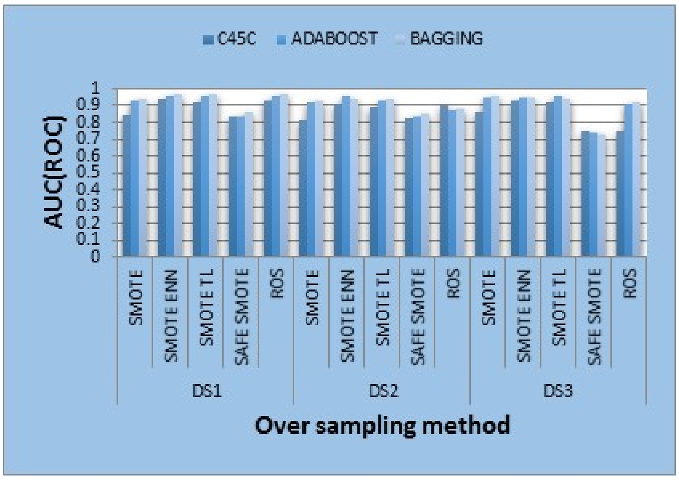

Fig. 2. AUC for oversampling methods

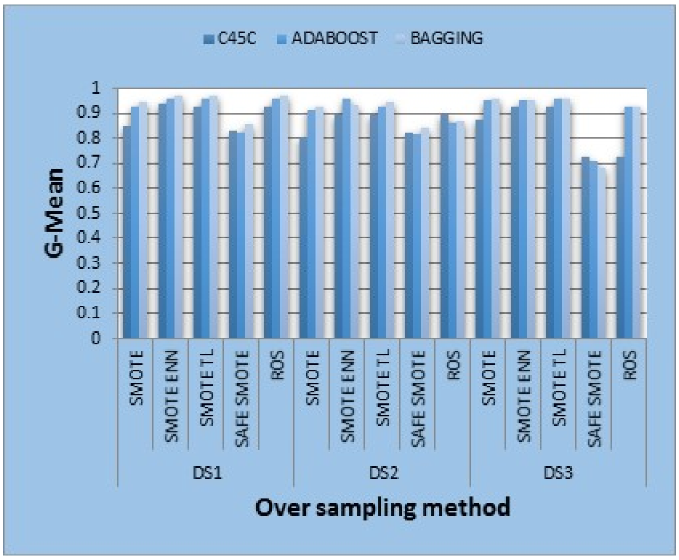

Fig. 3. G-Mean for oversampling method.

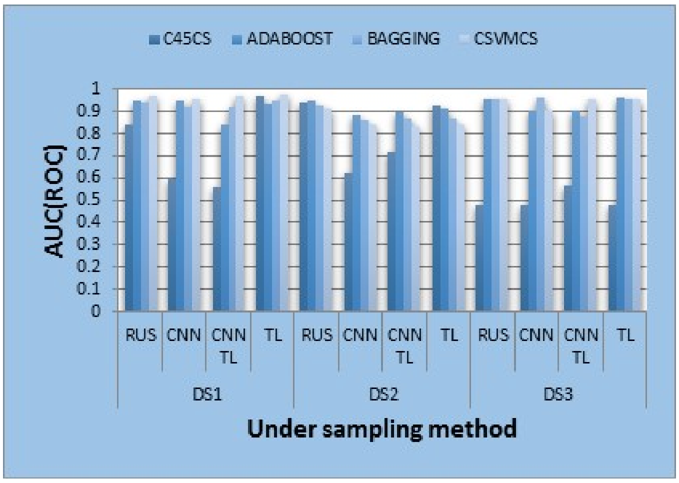

Fig. 4. AUC for undersampling methods.

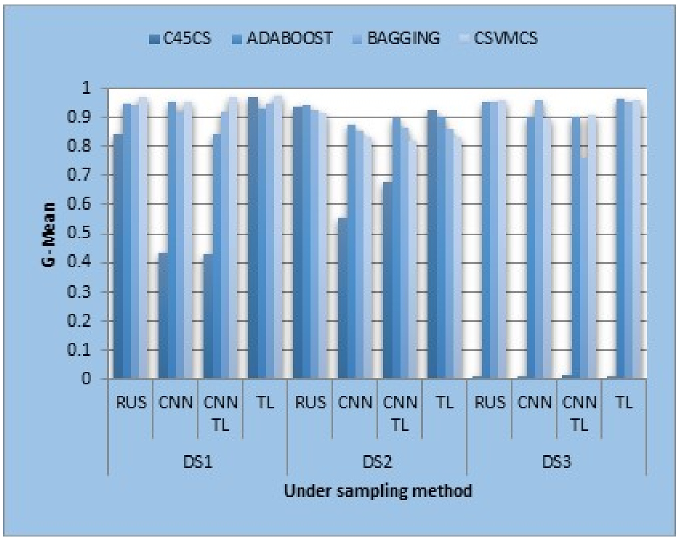

Fig. 5. G-Mean for under sampling method.

During experiments, we have observed that the time taken to apply a model containing under-sampling is very much lesser than the time taken to apply the same model, but by using oversampling. On closely studying all the above tables and bar graphs, we can find the sampling method which is producing a better result than the others. As we saw, in most of the cases, the SMOTE ENN has given better performance metric values in comparison with the others. While in the case of undersampling methods, the TL method has given better values of performance metrics as compared to the other methods.

V. Conclusion

In this paper, the effect of different sampling methods on the performance of the classifier, using credit card fraud date set with class imbalance is evaluated. The data used here consists of the twenty-eight principal components that are obtained by applying principal component analysis (PCA) on the real data, and the variables time, amount and class.

Three datasets are consisting of ten thousand, fifteen thousand and twenty thousand instances respectively. Five types of over-sampling methods, four types of under-sampling methods have been used. After that, some cost-sensitive classifiers and ensemble classifiers are applied to the data. Finally, the performance metrics are used to evaluate the performance of different methods.

On comparison of the classifiers which performed better with all the sampling methods are- the bagging classifier performed better than others when oversampling is used in the model, and the cost-sensitive support vector machines perform better in comparison to others in cases where undersampling is used. From the experiments that we have done, we can conclude that the SMOTE ENN oversampling method has given best performance than the others, while the TL undersampling method has given the best performance in comparison to others.

References

[1] N. S. Halvaiee and M. K. Akbari, "A novel model for credit card fraud detection using Artificial Immune Systems," *Applied Soft Computing Journal*, vol. 24, pp. 40–49, 2014.

[2] E. Michael and S. Pedro, "A survey of signature-based methods for financial fraud detection," *Computer and security*, vol. vol 28, no. 6, pp. 381–394.

[3] B. Adrian, "Detecting and Preventing Fraud with Data Analytics," *Procedia Economics and Finance*, vol. 32, no. 15, pp. 1827–1836, 2015.

[4] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[5] B. Zhu, B. Baesens, and K. L. M. Seppe, "An empirical comparison of techniques for the class imbalance problem in churn prediction," *Information Sciences*, vol. 408, pp. 84–99, 2017.

[6] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating Probability with Undersampling for Unbalanced Classification," in *In IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 2015, pp. 159–166.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[8] G. E. a. P. a. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training

data," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, p. 20, 2004.

[9] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the imbalanced class problem," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5476 LNAI, pp. 475–482, 2009.

[10] P. Hart, "The condensed nearest neighbor rule (Corresp.)," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 515–516, 1968.

[11] I. Tomek, "Two Modifications of CNN," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 11, pp. 769–772, 1976.

[12] B. Krishnapuram, S. Yu, and R. B. Rao, *Cost-Sensitive Machine Learning*. CRC Press, 2011.

[13] Y. Sahin, S. Bulkan, and E. Duman, "A cost-sensitive decision tree approach for fraud detection," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5916–5923, 2013.

[14] D. Yuanhong, C. Hongchang, and P. Tao, "Cost-Sensitive Support Vector Machine Based on Weighted Attribute," in *International Forum on Information Technology and Applications (IFITA'09)*, 2009, pp. 690–692.

[15] H. Falaki, "AdaBoost Algorithm," *Startrinity*. [Online]. Available: http://startrinity.com/VideoRecognition/Resources/Adaboost/boosting algori%0Athm 2.pdf.

[16] J. R. Quinlan, "Bagging, boosting, and C4.5," in *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 2006, vol. 5, pp. 725–730.

[17] S. Russell, "sample complexity," p. 122, 1994.

[18] M. Al Helal, M. S. Haydar, S. Al, and M. Mostafa, "Algorithms Efficiency Measurement on Imbalanced Data using Geometric Mean and Cross-Validation," in *International Workshop on Computational Intelligence (IWCI)*, 2016, pp. 110–114.

[19] J. West and M. Bhattacharya, "Some Experimental Issues in Financial Fraud Mining," *Procedia - Procedia Computer Science*, vol. 80, pp. 1734–1744, 2016.

[20] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.