Modern Applied Science; Vol. 14, No. 7; 2020
ISSN 1913-1844 E-ISSN 1913-1852
Published by Canadian Center of Science and Education

Analysis of Resampling Techniques on Predictive Performance of Credit Card Classification

Maira Anis[1], Mohsin Ali[2], Shahid Aslam Mirza[2] & Malik Mamoon Munir[3]

[1]Department of Management Science, Faculty of Business Administration, Muhammad Ali Jinnah University, Karachi, Pakistan

[2]Balanced Scorecard Institute Pakistan, Karachi, Pakistan

[3]Department of Management Science, Bahria University Islamabad, Pakistan

Correspondence: Dr Maira Anis, Department of Management Science, Faculty of Business Administration, Muhammad Ali Jinnah University, Karachi, Pakistan

Received: May 20, 2020 Accepted: June 22, 2020 Online Published: June 23, 2020

doi:10.5539/mas.v14n7p92 URL: https://doi.org/10.5539/mas.v14n7p92

Abstract

Credit card fraud detection has been a very demanding research area due to its huge financial implications and rampant applications in almost every area of life. Credit card fraud datasets are naturally imbalanced by having more legitimate transaction in comparison to the fraudulent transactions. Literature represents numerous studies that are aimed to balance the skewed datasets. There are two major techniques of resampling in balancing these sets i.e. under-sampling and oversampling. However both under-sampling and oversampling techniques suffer from their own set of problems that can seriously affect the performance of classifiers that have been inducted for credit card studies in the past. Thus to accelerate detection of credit card fraud, it is very important to implement the strategy that could possibly provide better predictive performance. This paper attempts to find out what resampling technique can work best under different skewed distributions for the domain of credit card fraud detection.

Keywords: credit card fraud detection, supervised classification, resampling techniques, class imbalance learning

1. Introduction

Over recent years the rampant application of credit card has led many losses to financial institutions and other recipient organizations. This has made detecting credit card fraud a hard challenge for concerned authorities. Credit card is considered as an easy fraud target because the fraudsters can gain a lot of money in a very short period of time and with less risk; as the fraud is detected after many days (Zareapoor & Shamsolmoali, 2015). Credit card fraud detection has been a very arduous research area due to the losses generated by these plastic gadgets. In 2015, according to “The Nilson Report”(Neilson, 2012), only in United States of America (USA) the credit card fraud has increased to 12.75 cents for every $100 annually and it contributes 21.4% of the total fraud losses across the world. In another study (Stolfo et al., 1997), it is revealed that worldwide 40% of the total financial losses are only generated by the credit cards alone. To reduce the losses to minimum by the stolen or misused cards, it is very necessary to block these cards as quickly as possible. Fraudsters use a lot of techniques in attempting frauds and always look for the sensitive information related to the card stolen. In this regard the financial institutions also adopt number of solutions to combat fraud. These techniques usually involve the process of classifying transaction either to fraud or non-fraud.

Credit card fraud datasets have been found to be naturally skewed (He & Garcia, 2009) which mean that these datasets have more legitimate transactions than the fraudulent ones. These imbalances between the majority and minority classes bias the classifiers to the majority class and misclassify the instances of the class that has less representation in the data. Usually in the classification process, class with less representation is more important than the other classes (Rahman & Davis, 2013). In credit card fraud detection, the instances belonging to the minority class i.e. the fraudulent transactions are of prime interest. Classification algorithms utilized in detection of credit card fraud are often overwhelmed with the majority class (Tremblay et al., 2007, Anis & Ali, 2017 and Shen et al., 2007) leading to bad predictive performance for the minority class. To increase the prediction rate of minority class, a lot of studies in literature have given numerous resampling techniques. However there are two basic techniques which are followed widely. These include Over-Sampling and Under-Sampling or combination of both Over-Sampling and Under-Sampling: which is called Hybrid Sampling.

- Under-Sampling: it removes the majority samples to the desired level of imbalance.
- Over-Sampling: generate new minority samples to the desired level of imbalance.
- Hybrid Sampling: This implements both over-Sampling and under-Sampling techniques until we reach the desired level of imbalance.

There is wide range of resampling techniques that are implemented for credit card frauds but among them we have selected three mostly used resampling techniques. In this study, resampling techniques utilized and compared are Random Over-Sampling (ROS), Random Under-Sampling (RUS) and SMOTE. In this study we have implemented these techniques to balance the datasets. These resampling techniques provide varied predictive performance for different classifiers. Thus we aim to explore and analyze these techniques for classification algorithms that have been widely implemented for credit card fraud detection.

An exhaustive list of classification algorithms have been inducted for the credit card studies. However there are some algorithms that have been extensively used for this purpose. For example Shen et al applied Decision Tree, Logistic Regression and Neural Network to analyze their performance for credit card fraud detection (Shen et al., 2007). Anis et al implemented the family of DTs for different levels of imbalance for credit card fraud (Anis et al., 2015). Brown and Mues also analyzed different classification models for a set of skewed levels to check predictive performance for the minority class in credit scoring (Brown & Mues, 2012). Peng et al ranked the most implemented classification algorithms for the credit card frauds (Peng et al., 2011). Considering the studies that have been specifically formulated to find the best classification algorithms, we utilized two famous and widely implemented algorithms that include, Decision Trees and Support Vector Machine. West and Bhattacharya presented a comprehensive literature review of financial fraud detection studies and found that Logistic Regression, Bayesian Belief Network, Support Vector Machines, Logistic Regression, Neural Network are the algorithms that perform optimally for credit card fraud (West & Bhattacharya, 2016 and Zhang &Zhou, 2004).

2. Methods

As explained in section 1, objective of this paper is to explore resampling strategies in balancing the imbalanced datasets for the classification algorithms. Thus we will provide a brief overview of the classification algorithms and notation for the problem statement and the resampling strategies and the evaluation metrics that have been implemented for this study.

2.1 Classification Algorithms

Decision Trees: Decision Tree is a technique of classifying data by generating a tree like structure. This tree has internal nodes that represent binary choices for each attribute whereas the branches of the tree symbolize the outcomes of that choice (Breiman, 2001). These nodes are created in such a way that the samples could be traversed using them. Decision Trees have many types e.g. Classification and Regression Trees (CART), J48 and Random Forest etc. Among the family of DT's Random Forest (RF) or decision forest is the widely used classification tree (Breiman, 2001). RF is collection of trees that are created to minimize the risk of over training the samples and to avoid the instability with in a single tree (Bhattacharyya et al., 2011). Another technique in DTs is pruning: which is used to reduce overfitting. Pruning removes the nodes of a DT without affecting the overall performance of a tree. Pruning also makes RF robust to noise and over training of samples. In RF each tree is created independently with little complexity and thus it requires tuning of only two parameters that include number of attributes and number of trees at each node. This process makes the generation of RF very simple (Bhattacharyya et al., 2011).

Support Vector Machine: Support Vector Machine (SVM) was developed by Vapnik, 1995. It is a classification technique of mapping linear functions to higher dimensional space. This enables a nonlinear complex classification problem to be solved linearly with minimum computational complexity. SVM uses a kernel function for the transformation of data to high dimensional space. Kernel function is defined as a linear mapping between the data and a high dimensional space. Mathematically it is given by:

$$k(x_1, x_2) = \langle \varphi(x_1), \varphi(x_2) \rangle$$

Where $\varphi: X \rightarrow H$ represents a mapping from function X to the higher dimensional space H. After the implementation of kernel function a hyper plane is generated to classify the data points to their respective classes and is defined as:

$$w.\emptyset(x) + b = 0$$

This hyper plane is constructed to have maximum separation between the instances of both the classes. Thus the final classification of SVM can be defined as:

$$\sum_{i=1}^{n} \propto_i y_i k(x_i, x) + b = 0$$

3. Problem Statement and Notations

This section will define the problem statement and the necessary notations that have been utilized in resampling and classification of credit card fraud datasets. As the credit card fraud transaction has to be classified to either legitimate or fraudulent, therefore we will consider a binary classification problem. Consider a data set D having m elements. Then $\mathrm{D} = (\mathrm{x_i}, \mathrm{y_i})_{\mathrm{i}=1}^{\mathrm{m}}$. Where $\mathrm{x_i}$ is a set of d-dimensional transactions and $\mathrm{y_i}$gives the labels i.e. $\mathrm{y_i} = \{0,1\}$. Here 0 & 1 represents majority and minority classes respectively. Let

$\mathrm{D_{maj}} = \{(\mathrm{x_i}, \mathrm{y_i}) \in \mathrm{D} | \mathrm{y_i} = 0\}$ &

$\mathrm{D_{min}} = \{(\mathrm{x_i}, \mathrm{y_i}) \in \mathrm{D} | \mathrm{y_i} = 1\}$.

As credit card fraud datasets are imbalanced where this imbalance can be described by defining an imbalance ratio i.e.

$$\mathrm{IR(D)} = \frac{\mathrm{D_{maj}}}{\mathrm{D_{min}}}$$

It is worth to note that higher ratio of $\mathrm{IR(D)}$ will give more skewed dataset. Thus our aim is to resample the dataset by lowering the $\mathrm{IR(D)}$. For $\mathrm{IR(D)} = 1$ we will acquire a fully balanced dataset. Every standard classification problem is modeled on some training data whereas the modeled is verified using the test data. Here we assume that D is the training data that needs to be resampled. Learning a classifier from imbalanced training set D can be done in two stages. In the first phase the dataset D is resampled such that a desired imbalance ratio $\mathrm{IR(r(D)}$ is achiever whereas $\mathrm{IR(r(D))} < \mathrm{IR(D)}$. This is performed by dropping majority transactions or by adding new minority samples that will be generated synthetically. After performing the resampling a standard classification function C is learned on resampled dataset $\mathrm{r(D)}$ to generate a model $\mathrm{C_{r(D)}}$that maps all the instances in m- dimensional space to the target set $\{0,1\}$ i.e. $\mathrm{C_{r(D)}}: \mathbb{R}^{\mathrm{m}} \rightarrow \{0,1\}$.

Next step is to validate the model $\mathrm{C_{r(D)}}$ by checking its performance on the test set $\mathrm{D_{test}}$using classifier C. For this purpose performance of any classifier is determined using performance metrics P for which the input is the trained model on the resampled dataset $\mathrm{C_{r(D)}}$ and $\mathrm{D_{test}}$ to produce better classification metrics. Higher values of these metrics give a better predictive model. In order to find the performance of the parameter r on the classifier C, k-fold cross validation is implemented during the training phase.

4. Resampling Techniques

Each resampling method r considered in this study, will follow a schematic way given below.

It will take input of training dataset for the resampling. A resampling multiplier l will be adjusted so that $\mathrm{IR}\big(\mathrm{r(D)}\big) = \frac{1}{\mathrm{l}}\ \mathrm{IR(D)}$ where, $\mathrm{l} > 1$. l is called the resampling multiplier that is used to regulate the amount of resampling.

Training dataset will be modified by adding new minority samples (oversampling) or by reducing the majority samples (undersampling). This will be done according to the method implemented for resampling.

Finally we get a resampled dataset $\mathrm{r(D)}$ that can be classified using classifier C where $\mathrm{IR}\big(\mathrm{r(D)}\big) \leq \mathrm{IR(D)}$.

Now we explain the resampling techniques we will use in this paper.

4.1 Random Undersampling

Random Under-Sampling is an effective technique that tends to eliminate the majority samples from the training data. A number of studies point towards the effectiveness of this sampling technique. In a study presented by Liu, it was found that by reduction of majority samples in large number can bring significant savings in terms of the training time and memory that is required in building a training model (Liu, 2004). However, randomly eliminating majority instances by great number can lead to drop useful information necessary in building a model that can detect more minority samples. It was suggested in Ganganwar, that RUS procedure should be performed on larger datasets idyllically as the larger datasets are acceptably redundant in majority samples, thus most of the data to be discarded is redundant in nature (Ganganwar, 2012). In another study, it revealed RUS as the naivest and most frequently used resampling technique. Major drawback faced by RUS technique is that the amount of information withdrawn from the training data cannot be controlled (Krishnaveni & Rani, 2011). Despite the fact, that this technique can affect the classifiers performance, RUS have been considered as the most effective technique and can outperform certain other sophisticated techniques (Wang et al., 2008).

RUS is performed until both the classes (i.e. majority and minority) have same number of samples. Also RUS does not take in to account any additional parameters. For this study, a random subset of $\mathrm{D_{maj}}$ with

$$|\mathrm{D_{maj}}|(\frac{\mathrm{l}-1}{\mathrm{l}})$$

samples that will be withdrawn. All the samples in $\mathrm{D_{maj}}$ have equal probability to be selected for the process of under-sampling.

4.2 Random Oversampling

Random Over-Sampling (ROS) inclines to increase the number of samples of the minority class. ROS until they represent a balance number of samples with respect to the majority class samples in the training data. A detailed analysis of over-sampling is given by Chawla in which the importance of over-sampling has been emphasized (Chawla et al., 2002). ROS retains the existing information of the dataset in contrast to RUS. However, the shortcomings of ROS were marked and significantly include need of large memory and longer time in training the model because of greater number of samples for both classes (Wang et al., 2008). In another study, it was further pointed out that ROS create an issue of overfitting due to replication of minority instances and thus the model cannot be generalized to the new data (Ganganwar, 2012).

Despite the fact that this technique holds certain limitations, Liu insisted the use of ROS as a very effective procedure of resampling (Liu, 2004). In his study, it was suggested to generate new minority instances from the existing training data rather generating new instances from the new training set could possibly bias the process of random selection of instances.

In this study, minority sample will be randomly generated until they become equal in number with the majority samples. For this purpose, $|\mathrm{D_{min}}|(\mathrm{l}-1)$ minority samples are added to the training set.

4.3 SMOTE

SMOTE stands for Synthetic Minority OverSampling Technique. This novel technique was presented by Chawla et al., 2002. SMOTE mainly creates a new sample by interpolation of existing minority samples that lie together. For any original sample $\mathrm{x_i}$, it randomly selects one or more k nearest neighbors of $\mathrm{x_i}$ and performs interpolation of the existing sample and its neighbor and creates a new sample. More specifically, it follows the subsequent procedure in creating new samples. SMOTE takes the difference between $\mathrm{x_i}$ and its nearest neighbor, this difference is multiplied by a random number between 0 and 1. Finally this is added to the original sample $\mathrm{x_i}$ to get a new sample $\mathrm{x_{new}}$. This technique forces the decision region of the minority class towards the majority space that can effectively reduce the problem of overfitting. Although SMOTE significantly improves the performance of minority class, it hinders the performance of classifiers by assigning the same sampling rate to each neighboring instance of $\mathrm{x_i}$. To overcome this problem, certain SMOTE based studies have been proposed to assign different weightings to the neighboring minority class instances for $\mathrm{x_i}$, e.g. (Lu & Ju, 2011 and Ngai et al., 2011). SMOTE uses an additional parameter k for defining the sampling rate. FOR this study we have implemented SMOTE for k=5.

Following procedure is adopted in synthetically generating new minority samples.

1. Initialize a new set $D_{new} = \varphi$.
2. To generate new samples, following steps have been repeated $|D_{min}|(l-1)$times:

i) Randomly select $x_i \in D_{min}$.

ii) Find k nearest neighbors of x_i. Randomly choose any nearest neighbor and call it x_j.

iii) Interpolate these two samples in the following way to find new sample for x_i:

$x_{new} = x_i + \alpha(x_i - x_j) \quad \alpha \epsilon [0\ 1]$

iv) Label all new samples as the minority class samples and add to D_{new}

3. Add the newly generated samples to the set D i.e. $D = D_{new} + D_{min} + D_{maj}$

4.4 Performance Measures

The most common measure in classification is accuracy. However, high accuracy does not imply that all the fraudulent transactions have been classified correctly. Cost of misclassifying fraudulent transaction is far greater than misclassifying legitimate transaction in credit card fraud detection. As accuracy of any classifier cannot characterize the performance for the minority class, it is considered to be a biased metric. Other than accuracy, there have been other measures developed by the data mining community. An assortment of these metrics is based on a confusion matrix that is illustrated in Table 1. Confusion matrix is a 2x2 matrix with 4 elements described under:

Table 1. Confusion Matrix

Actual Class	Predicted Class	
	Positive	Negative
Positive	TP (Hit/Fraud catch)	FN (Miss)
Negative	FP (False Alarm)	TN (Normal)

True Positive (TP) represents the accuracy for the fraudulent (positive) examples. It explains how many positive examples have been labeled correctly. TP rate is also called sensitivity or recall. Similarly True Negative (TN) rate defines how many legitimate examples are classified correctly. It is also called specificity. False Negative (FN) detects the number of positive examples classified as negative where as True Negative (TN) represents negative examples predicted negative. Performance measures utilized in this study are based on the elements of the confusion matrix. These evaluation metrics are defined as follows.

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

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

$$F - measure = \frac{(1 + \beta^2)Precision \times Recall}{(\beta^2 . Precision + Recall)}$$

Whereas $\beta = 1$ and

$$Precision = \frac{TP}{TP + FP}$$

5. Experimentation

Credit card fraud detection is an area of fraud detection that is more explored during recent years. The methods adopted to detect credit card fraud support an auto detection of fraudulent behavior among the given transaction. However, there are some constraints t this domain follows either naturally or due to some restrictions imposed by the financial institutions. Firstly the credit card datasets are heavily skewed (Juszczak et al., 2008 and He et al., 2008)and the real datasets are mostly not provided by the financial institutions (Lu & Ju, 2011 and Ngai et al., 2011) due to privacy concerns of the customers. Also the datasets available have very low number of samples which becomes the cause of not learning all the rules by the classifier. In this paper we are utilizing 3 datasets. 2 datasets, German Credit Card and Australian Credit Card datasets have been taken from UCI repository (Asuncion & Newman, 2010). These datasets have been implemented in most of the studies (Peng et al., 2011, West & Bhattacharya, 2016 and Li et al., 2013). Third dataset, Give Me Some Credit (GMSC) have been obtained from kaggle repository that was used for a competition. All the dataset utilized in this paper contain different ratios of fraud and non-fraud transactions (www.kaggle.com). For all datasets, 70% of the data is kept for the training and validation while 30% is used for the testing purpose. As the credit card datasets are extremely skewed, each training dataset, D is altered to have imbalance ratios of $\mathrm{D_i}$ where $\mathrm{i} = 1,2,3,4$. These datasets contains different ratios of fraud transactions i.e. 5%, 10%, 20% and 30%. Each $\mathrm{D_i}$ is further bifurcated to its corresponding majority and minority class instances se i.e. $\mathrm{D_{imaj}}$ and $\mathrm{D_{imin}}$ for application of corresponding resampling method r.

For the classification of three datasets, we selected two classifiers that have been explained in section 1. The classifiers SVM and DT are executed using default parameters. For SVM radial kernel was used. Classification is implemented using 10-fold cross-validation. This means that during the training phase the dataset have been divided in to 10 equal parts. Among 10 parts, 9 have been used to build model while 1 part of the training data is used to validate the model. This process is repeated until every part of the training data is utilized for the validation process.

6. Results and Discussion

In this paper, we have analyzed three mostly implemented resampling techniques for imbalanced datasets along with no resampling. Results for the three datasets using two classifiers are presented from Figure 1 to Figure 3. For all the datasets, we utilized four imbalanced distributions and the results of resampling techniques have been given for three performance metrics defined in Section 2. Results for each resampling technique are plotted separately against different imbalance ratios for respective evaluation metrics.

For Australian dataset, results are presented in Figure 1. Although "no resampling" has not achieved good performance but in some cases it has performed optimally than all the other resampling techniques for both classifiers. SMOTE and ROS approximately performed equivalently for all the distributions whereas RUS performed well only for TP rate for both classifiers. Similarly for German dataset, results are depicted by Figure 2. RUS performs optimally for all the distributions. It has been perceived that when data is not extremely skewed i.e. for 70:30 and 80:20, SMOTE gives better results but for the extreme distribution datasets i.e. 90:10 and 95:05, it fails to achieve better scores like previous datasets for both the classifiers. Here also no resampling achieved insignificant performance for both classifiers.

For the dataset GMSC, results are depicted in Figure 3. It can be seen that for both the classifiers, as the imbalance ratio is increased, no resampling either fails to provide better evaluation metrics or the performance get worse with increasing ratio of imbalance. However for other resampling techniques, SMOTE performs well for all the metrics. RUS also provide better results for highly imbalanced datasets of GMSC. Similar behavior is observed for ROS.

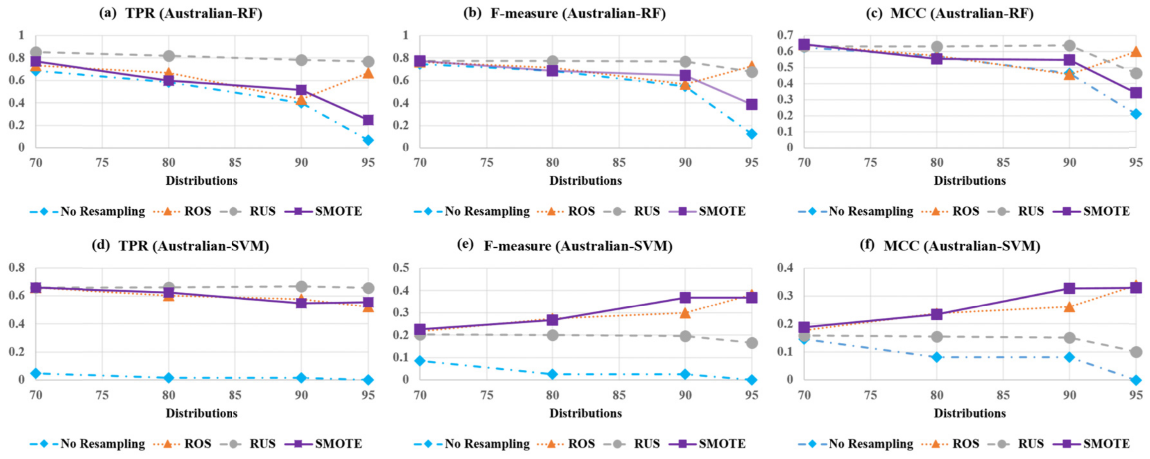

Figure 1. Performance of classifiers for Australian dataset in terms of TP rate, F-measure and MCC

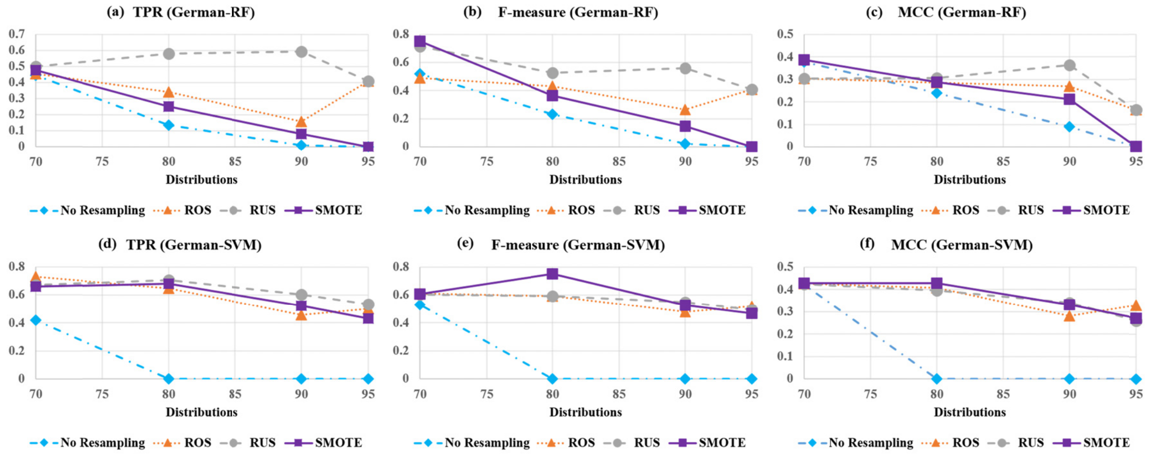

Figure 2. Performance of classifiers for German dataset in terms of TP rate, F-measure and MCC

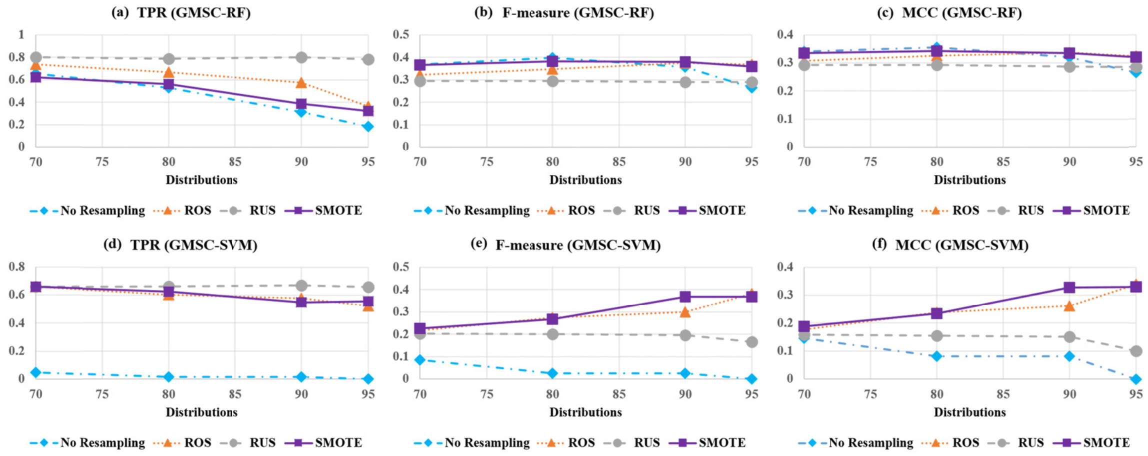

Figure 3. Performance of classifiers for GMSC dataset in terms of TP rate, F-measure and MCC

In summation for all the datasets, the results do not provide dominance of any resampling technique over another. However it has been noticed that every resampling technique is giving highest performance score for certain evaluation metric e.g. in almost all the cases TP rate has been achieved by the RUS approach. Similarly for F-measure SMOTE and ROS out rule RUS whereas no resampling fails to provide better performance score in all datasets except for some measures. This implies to choose resampling techniques very carefully else the performance of the classifier will be adversely affected as for some cases the resampling techniques have performed worse than even no resampling. This shows that not all imbalanced datasets require resampling techniques.

Quality of datasets is also an important factor affecting the predictive performance of classifiers. Performance of resampling technique is highly dependent on the classifier selected for the prediction e.g. for German dataset, SMOTE has performed equivalently to no resampling for RF when the dataset becomes extremely skewed whereas for the same dataset, SMOTE achieved best evaluation metrics for SVM. In certain cases it has been also observed that at extreme distribution the resampling techniques are performing very well however, the same techniques are not presenting optimal performance for the other dataset at the same distribution. Thus it is very important to know to what extent these resampling techniques can be implemented because balancing the datasets does not guarantee to have good metrics of prediction.

7. Conclusion

This study has looked at the comparison of widely used traditional resampling techniques and no resampling for credit card fraud detection. For this purpose, performance of three credit card datasets with varying levels of imbalance was assessed using three evaluation metrics. After the analysis of these techniques it is concluded that generally resampling improves the performance of imbalanced datasets in most of the cases. It is necessary to know to what extent the data can be resampled for different levels of imbalance. Balancing the dataset does not provide an optimal performance in all cases and can deteriorate the performance of imbalanced datasets. Moreover there cannot be any unanimous choice for resampling techniques for credit card imbalanced datasets and very unexpectedly there exist some cases for which no resampling can be a better choice. Thus in order to improve the classification results for imbalanced credit card datasets one needs to specify an optimal resampling technique and should also consider no resampling. It is also important to know what threshold of resampling can give better scores of evaluation metrics as balancing the dataset is not a good choice in all cases.

In our future work, we aim to compare these balancing techniques with different thresholds of resampling to assess the performance of imbalanced credit card datasets.

References

Anis. M. & Ali. M. (2017). A Novel Similarity Based Undersampling of Imbalanced Datasets. *International Journal of Computer Science and Information Security, 15*(1), 134:139.

Anis. M., Ali. M. & Yadav. A. (2015). A Comparative Study of Decision Tree Algorithms for Class Imbalanced Learning in Credit Card Fraud Detection. *International Journal of Economics, Commerce and Management, III*(12), 86-102.

Asuncion. A. & Newman. D. J. (2010). UCI Machine Learning Repository. In: School of Information and

Computer Science, University of California, Irvine, CA.

Bhattacharyya. S., Jha. S., Tharakunnel. K. & Westland. J. C. (2011). *Data mining for credit card fraud: A comparative study, 50*, 602-613. Decision Support Systems. https://doi.org/10.1016/ j.dss.2010.08.008

Breiman. L. (2001). *Random Forests. Machine Learning, 45*(1), 5-32. https://doi.org/10.1023/A: 101093404324

Brown. I. & Mues. C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications, 39*, 3446–3453. https://doi.org/10.1016/j.eswa.2011.09.033

Chawla. N. V., Bowyer. K. W., Hall. L.O. & Kegelmeyer. W. P. (2002). SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research, 16*, 321-357. https://doi.org/10.1613/jair.953

Ganganwar, V. (2012). An Overview of Classification Algorithms for Imbalanced Datasets. *International Journal of Emerging Technology and Advanced Engineering, 2*(4), 42-47.

He. H. & Garcia. E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 21*(9), 1263-1284. https://doi.org/10.1109/TKDE.2008.239

He. H., Bai. Y., Garcia. E. A. & Li. S. (2008). ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. *IEEE International Joint Conference on Neural Networks,* 1322-13328. https://doi.org/10.1109/IJCNN.2008.4633969

http://www.kaggle.com/c/GiveMeSomeCredit

Juszczak. P., Adams. N. M., Hand. D. J., Whitrow. C. & Weston. D. J. (2008). Off the peg and bespoke classifiers for fraud detection. *Computational Statistics and Data Analysis, 52*(9), 4521-4532. https://doi.org/10.1016/j.csda.2008.03.014

Krishnaveni. C. V. & Rani. T. S. (2011). On the Classification of Imbalanced Datasets. *International Journal of Computer Science and Technology, 2*(1), 145-148. DOI:10.13140/RG.2.2.14964.24961

Li. H., Zou. P., Wang. X. & Xia. R. (2013). A New Combination Sampling Method for Imbalanced Data. Proceedings of 2013 Chinese Intelligent Automation Conference, Lecture Notes in Electrical Engineering 256. Springer-Verlag Berlin Heidelberg 2013. https://doi.org/10.1007/978-3-642-38466-0_61

Liu. A. Y. (2004). The Effect of Oversampling and Undersampling on Classifying Imbalanced Text Datasets. (2012). The University of Texas. DOI:10.1.1.101.5878&rep=rep1&type=pdf

Lu. Q. & Ju. C. (2011). Research on Credit Card Fraud Detection Model Based on Class Weighted Support Vector Machine. *Journal of Convergence Information Technology, 6*(1), 62-68. https://doi.org/10.4156/jcit.vol6.issue1.8

Neilson (2012). Global Cards — 2011. The Nielsen Report, April 2012 (Issue 992), Carpinteria, CA, USA. www.prweb.com/releases/2013/8

Ngai. E. W. T., Hu. Y., Wong. Y. H., Chen. Y. & Sun. X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems, 50*(3), 559-569. https:/doi.org/10.1016/j.dss. 2010.08.006

Peng. Y., Wang. G., Gang. K. & Shi. Y. (2011). An empirical study of classification algorithm evaluation for financial risk prediction. *Applied Soft Computing Journal, 11*(2), 2906–2915. https://doi.org/10.1016/j.asoc.2010.11.028

Rahman. M. & Davis. D. N. (2013). Addressing the Class Imbalance Problems in Medical Datasets. *International Journal of Machine Learning and Computing, 3*(2), 224-228. https://doi.org/10.7763/IJMLC.2013.V3.307

Shen. A., Tong. R. & Deng. Y. (2007). Application of Classification Models on Credit Card Fraud Detection. *IEEE International Conference on Service Systems and Service Management,* 1-4, 2007. https://doi.org/10.1109/ICSSSM.2007.4280163

Stolfo. S. J., Fan. D. W., Lee. W., Prodromidis. A. L. & Chan. P. K. Credit Card Fraud Detection Using Meta-Learning: Issues and Initial Results. In AAAI-97 Workshop on AI Methods in Fraud and Risk Management, 1997.

Tremblay. M., Pater. R., Zavoda. F., Valiquette. D. & Simard. G. (2007). Accurate Fault-Location Technique based on Distributed Power-Quality Measurements. 19th International Conference on Electricity

Distribution.

Vapnik. V. (1995). The Nature of Statistical Learning Theory. Springer, New York, NY, USA. https://doi.org/10.1007/978-1-4757-2440-0

Wang. S., Tino. P. & Kaban. A. (2008). Class Imbalance Learning. University of Birmingham.

West. J. & Bhattacharya. M. (2016). Intelligent financial fraud detection: A comprehensive review. *Computers & Security, 57*, 47-66. https://doi.org/10.1016/j.cose.2015.09.005

Zareapoor. M. & Shamsolmoali. P. (2015). Application of Credit Card Fraud Detection: Based on Bagging Ensemble Classifier. *Procedia Computer Science, 48*, 679–686. https://doi.org/10.1016/j.procs.2015.04.201

Zhang. D. & Zhou. L. (2004). Discovering golden nuggets: data mining in financial application. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 34*, 513-22. https://doi.org/10.1109/TSMCC.2004.829279

Copyrights