Southeast University Journal of Computing Sciences, Volume 01, No 01, June 2021

Credit Card Fraud Detection Employing Various Sampling Techniques to Counter Class Imbalance

Apurba Hasan, Dept. of CSE, Southeast University, apurbahasan1994@gmail.com
Abdul Mueez , Dept. of CSE, Southeast University, abdul.mueez@seu.edu.bd

***Abstract*—With advancements in e-commerce and online-based payment systems, credit card fraud has also increased lately. Financial companies lose a huge amount of money as a result of it every year. Many machine learning approaches have been used to detect credit card fraud. However, there are a lotof challenges associated with it. The class imbalance problem is considered to be a major hurdle. It occurs when one class has considerably more observations than the other. The number of fraudulent transactions is usually so low comparedto legitimate transactions that classification algorithms fail to detect fraud transactions effectively. In this paper, multiple undersampling and oversampling techniques such as Synthetic Minority Oversampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), Borderline-SMOTE, SVMSMOTE havebeen used to overcome the class imbalance problem. Each of the techniques was explored rigorously with two machine learning algorithms - Random Forest and XGBoost. The performances of the algorithms have later been evaluated on the basis of precision, recall, and F-measure. According to the study, undersampling and oversampling techniques have a low success rate of detectingthe minority class when data is highly imbalanced. The precision and F-measure scores are very low in these approaches that lead to inaccurate detection of the minority class, fraud cases.**

***Keywords*—Credit Card Fraud, Class Imbalance, Random Forest, XGBoost, Random Undersampling, Nearmiss, SMOTE, ADASYN, Borderline-SMOTE, SVMSMOTE**

I. Introduction

Credit card fraud is a term used to denote any act of theft and fraud, occurred during payment while using a debit or credit card, that fraudsters gained accessed to by means of various techniques including but not limited to phishing, identity theft, etc. [1]. A large number of financial institutions and banks are interested in fraud detection as this crime costs them around USD 67 billion per year [2]. Malaysia accounted for around 320 million credit card transactions in 2011 and increased that to about 360 million in 2015 [3]. There are many research studies dedicated to credit card fraud detection. Many machine learning approaches have been used in various studies which include Random Forest (RF), Artificial Neural Network (ANN), Decision Tree (DT), AdaBoost and Majority Voting, Signature Analysis, Neural Data Mining, Feature Extraction. But most of the standard machine learning techniques fail to detect fraud transactions effectively due to the class imbalance problem. This problem causes machine learning algorithms to output a high accuracy score. However, a high accuracy score is not always an indication of the real world performance of these algorithms because valid transactions outweigh fraudulent ones in the real-world dataset. They can correctly classify the valid transaction but fail to detect the fraud ones as they end up treating them as noise in such a dataset.

In this paper, many undersampling and oversampling techniques are used to overcome the class imbalance problem. Different experiments have been performed to examine the performance of Random Forest and XGBboost on credit card fraud using Precision, Recall, and F-measure.

II. Related works

Paper [4] used Random Forest, Logistic Regression and Support Vector Machine to detect credit card fraud. The dataset used in this paper has credit card transaction information from January 2006 to January 2007. To evaluate the models performance various metrics such as accuracy, sensitivity, specificity, etc. were used. The overall result shows that Random Forest performs better than Support Vector Machine and Logistic Regression. A model named Artificial Immune Recognition Systems (AIRS) was proposed for credit card fraud detection in [5]. This model is a refinement to Artificial Immune Systems (AIS). To evaluate the model performance, False Positive Rate, Detection Rate and Hit Rate were used. The result indicated that detection rates improved by 25% while cost decreased by 85%. Paper [6] suggested a hybrid model for online fraud detection for video-on-demand systems. For recording the data, by including an artificial immune system based fraud detection, the proposed model intended to enhance the existing risk management pipeline (RMP). The AIS based model is the combination of two Artificial Immune System algorithms associated with behavior-based instruction detection classification and regression trees (CART). Paper [7] suggested the use of BLAST_SSAHA hybridization for spotting credit card fraud. To determine the similarity of incoming spending patterns, first a profile analyzer is used. Afterwards, the abnormal transactions tracked down by the profile analyzer are moved to a deviation analyzer (DA) for a likelihood of matching with past fraudulent behavior [7]. Artificial transactions were generated by a simulator to examine the performance of the proposed system there. True Positive Rate (TPR) and False positive Rate (FPR) ware

used as evaluation metrics. The results obtained point to a higher accuracy with the new approach. Simultaneously, the computational speed is quick enough to make credit card fraud identification viable [7]. A Hybrid model based on the combination of supervised and unsupervised techniques for detecting credit card fraud was proposed in [8]. The suggested model combines the behavioral model and the rule-based model. To enhance the data description boundary according to the account spending behavior one class classifier was used. The performance of the method was measured using TPR, FPR, TNR, FNR. The results indicate that the bank's rule- based techniques do not succeed in detecting the majorityof the fraudulent transactions that were identified with the hybrid technology [8]. Paper [9] proposed a Hidden Markov Model. A meta-classifier model was advanced in [10] which utilised Decision Tree, Naive Bayesian and k-nearest neighbor as base classifiers. Naive Bayesian was also chosen as the metaclassifier which used the predictions of the aforementioned algorithms as features in order to produce the final classifier. This work focused on 11 months of credit card transactions from a major Canadian bank. The meta classifier resulted in a 10% performance improvement in contrast to the bank's existing algorithm.

III. Overview of the Algorithms

A. Random Forest

Random Forest is known as an ensemble machine learning method which employs several decision trees as base classifiers [11].Each tree in a random forest method is trained with a random sample from the training data. The samples of the
dataset are drawn using a statistical method called bootstrap aggregation or bagging. The bagging technique uses some samples of the dataset multiple times in a single Decision Tree. The majority vote of the Decision Trees is then selected as the concluding result by the Random Forest Algorithm.

B. XGBoost

XGBoost [12] is an ensemble machine learning algorithm. The term XGBoost stands for Xtreme Gradient Boosting. Gradient boosting is a supervised machine learning algorithm where it combines multiple weak learners to develop a strong learner which is then used to make predictions. In Xtreme Gradient Boosting the base learners are Decision trees. The results of all base models are united by XGBoost to produce the eventual prediction.

IV. Methodology

In this section, at first, an overview of the dataset used in this paper is given. Then the evaluation metrics and the methods which are used to handle the class imbalance problem are discussed in details.

A. Dataset Description

The dataset [13] used in this research consists of transactions carried out using credit cards during the month of September in 2013 by cardholders in Europe. These transactions took place in two days. Out of 284,807 transactions, only 492 transactions are fraudulent, meaning that fraud cases account for only 0.172% of the dataset. This shows how imbalanced this dataset is. Moreover, there are 31 features in this dataset. These input variables of this dataset contain only numerical values. Features V1, V2, ...V28 are the main components of this dataset and they have been transformed via Principal Component Analysis (PCA) for security issues. The features, Time and Amount, are in their actual form as they have not been transformed with PCA. Time reports the elapsed time (in seconds) that was taken up during the first transaction and all the other corresponding transactions, while Amount contains the transaction amount.

B. Evaluation metric

In this paper, three evaluation metrics are used namely Precision, Recall, and F-measure. Precision shows how precisely the models detect fraudulent transactions. It is also known as positive predictive value.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

The Recall explains how many of the fraudulent transactions were predicted correctly by the models. It is also known as sensitivity.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

F-measure is the harmonic mean of Precision and Recall.

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

C. Resampling Method

In this paper, data undersampling and oversampling have been used to overcome the class imbalance problem. For undersampling, random undersampling and nearmiss were used. For oversampling, random oversampling, SMOTE, Adaptive Synthetic Sampling (ADASYN), Borderline-SMOTE and SVMSMOTE were used. In random undersampling, random samples from the majority class are selected from the original data sample. In nearmiss [14] KNN algorithm is used forundersampling the dataset. In random oversampling, random samples from the minority class are selected from the initial dataset with replacement. ADASYN algorithm employs the use of a density distribution to determine the number of gene samples that need to be generated for every minority data sample [15]. Only the borderline samples of the minority class are increased in frequency in Borderline-SMOTE. Most of the classification algorithms are made to attempt to learn the bor- derline of each class in the training process in order to obtain better prediction results [16]. In

SVMSMOTE, the borderline area is estimated by support vectors that were obtained after a standard SVM classifier had completed training on the originaltraining set [17].

V. Results

For experiment with actual dataset, 33% of the dataset is used for testing purposes and the rest of it for training purposes. In undersampling and oversampling methods, 10 fold cross-validation has been used. The parameters used for Random Forest and XGBoost are shown in Table I and Table II.

TABLE I
Parameters for Random Forest

Parameter Name	Value
n estimators	200
min samples split	100
min samples leaf	100
max features	20
max depth	5

TABLE II
Parameters for XGBoost

Parameter Name	Value
n estimators	100
subsample	0.9
min child weight	1
learning rate	0.2
max depth	8
gamma	0.1

A. Actual data

TABLE III
Performance on Actual Data

Evaluation metric	Random Forest	XGBoost
Precision	0.74	0.97
Recall	0.68	0.83
F-measure	0.71	0.90

XGBoost has a better fraud detection rate in this experiment with higher Recall (0.83) and F-measure (0.90) as results shown in Table III. The Precision (0.97) was also better in the case of XGBoost.

B. Random Undersample

TABLE IV
performance on random undersample

Evaluation metric	Random Forest	XGBoost
Precision	0.06	0.04
Recall	0.87	0.91
F-measure	0.11	0.08

XGBoost gave better Recall (0.91) than Random Forest (0.87) as results shown in Table IV. But the F-measure and Precision were low for both Random Forest (Precision=0.06, F-measure=0.11) and XGBoost (Precision=0.04, F-measure=0.08) as samples from the valid transaction predicted as fraud by the algorithms. This happened because random samples from the majority class are picked to balance the data. So a lot of information is lost which leads to low Precision and F-measure.

C. Nearmiss

TABLE V
Performance on Nearmiss

Evaluation metric	Random Forest	XGBoost
Precision	0.003	0.003
Recall	0.95	0.95
F-measure	0.007	0.007

Recall (0.95) was the same for Random Forest and XGBoostin the results shown in Table V. Precision (0.003) and F- measure (0.007) were very poor for both the algorithms because Nearmiss excludes data points that are close to each other.

D. Random Oversample

TABLE VI
performance on Random Oversample

Evaluation metric	Random Forest	XGBoost
Precision	0.18	0.76
Recall	0.84	0.76
F-measure	0.29	0.71

Random Forest gave better Recall (0.84) than XGBoost (0.76) in results of Table VI. But the F-measure and Precision were lower in Random Forest (Precision=0.18, F-measure=0.29) and XGBoost (Precision=0.76, F-measure=0.71) than the re- sult of the experiment with actual data shown in Table III. In random oversampling, an exact copy of the minority classes is generated to balance the data. This can make the algorithms misclassify the majority instances which in turn might have led to low Precision here.

E. SMOTE

TABLE VII
Performance on Smote

Evaluation metric	Random Forest	XGBoost
Precision	0.13	0.53
Recall	0.87	0.78
F-measure	0.22	0.58

Random Forest gave better Recall (0.87) than XGBoost (0.76) as indicated by the results shown in Table VII. F-measure and Precision were lower in Random Forest (Preci- sion=0.13, F-measure=0.22) and XGBoost (Precision=0.53, F- measure=0.58) than the result of the experiment with actual data shown in Table III. In SMOTE the synthetic samples that are generated might have overlapped with the majority instances which might have resulted in their misclassification. That's why the Precision score is low here.

F. ADASYN

Ramdom Forest gave better Recall (0.87) that XG- Boost (0.79) as results shown in Table VIII. F-measure and

TABLE VIII

PERFORMANCE ON ADASYN

Evaluation metric	Random Forest	XGBoost
Precision	0.14	0.40
Recall	0.87	0.79
F-measure	0.24	0.42

precision were low in both Random Forest (Preci- sion=0.14, F-measure=0.24) and XGBoost (Precision=0.40, F-measure=0.42) than the result of the experiment with actual data shown in Table III. ADASYN generates more synthetic data points where the density of the minority samples is relatively low and very few data points where the density of minority samples is high. Those low-density data points might be outliers which lead to the low performance of the models.

G. Borderline-SMOTE

TABLE IX

PERFORMANCE ON BORDERLINE-SMOTE

Evaluation metric	Random Forest	XGBoost
Precision	0.48	0.68
Recall	0.82	0.75
F-measure	0.57	0.66

Random Forest gave a better Recall (0.82) than XGBoost as results shown in Table IX. F-measure and Precision were better in both Random Forest (Precision=0.48, F-measure=0.57) and XGBoost (Precision=0.68, F-measure=0.66) than SMOTE shown in Table VII. This happened because Borderline-SMOTE selects data points that are being misclassified by the SMOTE method and gives more attention to them.

H. SVMSMOTE

TABLE X

PERFORMANCE ON SVMSMOTE

Evaluation metric	Random Forest	XGBoost
Precision	0.24	0.28
Recall	0.82	0.77
F-measure	0.28	0.28

Random Forest gave a better Recall score (0.82) than XGBoost (0.77). F-measure (0.28) was the same for both the algorithms as result shown in Table X. Precision was low in both Random Forest (0.24) and XGBoost (0.28). Borderline-SMOTE SVMSMOTE also selects data points of the minority class that were misclassified by SMOTE but it uses SVM to select them.

Figure 1 shows that Precision is high in actual data for Random Forest and XGBoost. But in Nearmiss and Random Undersampling it is very low. Figure 2 shows that Recall is high in Nearmiss for Random Forest and XGBoost. For the rest of the methods, Recall is very even. Fig 3 shows that F-measure is high in actual data for Random Forest and XGBoost. But in Nearmiss and Random Undersampling it is very low.

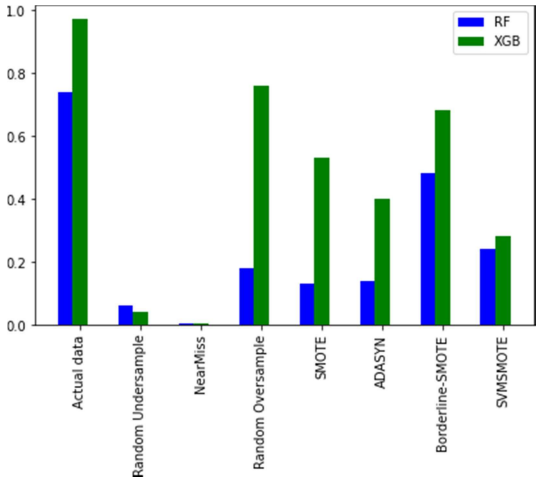

Fig. 1. Precision for Random Forest and XGBoost.

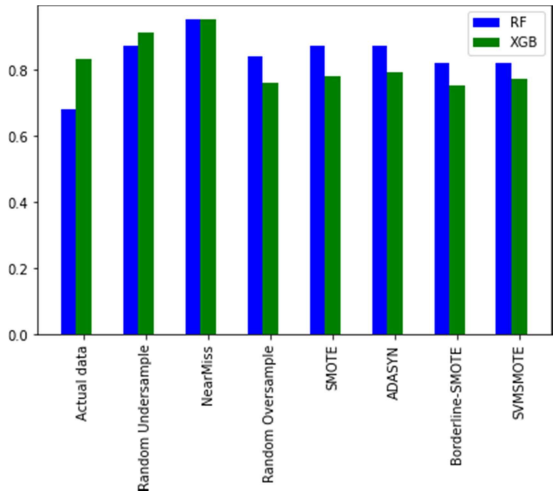

Fig. 2. Recall for Random Forest and XGBoost.

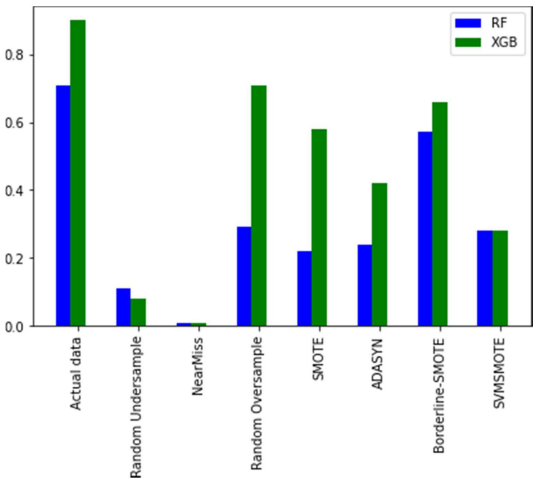

Fig. 3. F-measure for Random Forest and XGBoost.

VI. CONCLUSION

Credit Card Fraud detection is a very challenging task. Individuals and many business organizations bear a huge amount of financial loss as a result of it. Hence, machine learning algorithms have been used extensively to detect fraudulent transactions. This paper aims to compare the performance of Random Forest and XGBoost in credit card fraud detection after a dataset has undergone various sampling procedures. Undersampling and oversampling techniques have been used to overcome the class imbalance issue. To judge the performance of the models, Precision, Recall, and F-measure have been used. The results of these experiments showed that Random Forest had a better Recall score in data oversampling methods than XGBoost. XGBoost had a better recall in actual and random undersampled data and it was the same in nearmiss for both Random Forest and XGBoost. But on the other hand, Precision was higher for all the undersampling and oversampling methods except nearmiss. The F-measure was higher for XGBoost in most of the oversampling methods except SVMSMOTE. It was higher for Random Forest in actual, random unsampled and nearmiss data.

REFERENCES

[1] D. Devi, S. K. Biswas, and B. Purkayastha, "A cost-sensitive weighted random forest technique for credit card fraud detection," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2019, pp. 1–6.

[2] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. Zeineddine, "An experimental study with imbalanced classification approachesfor credit card fraud detection," IEEE Access, vol. 7, pp. 93 010–93 022, 2019.

[3] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, "Credit card fraud detection using adaboost and majority voting," IEEE access, vol. 6, pp. 14 277–14 284, 2018.

[4] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," Decision Support Systems, vol. 50, no. 3, pp. 602–613, 2011.

[5] N. S. Halvaiee and M. K. Akbari, "A novel model for credit card fraud detection using artificial immune systems," *Applied soft computing*, vol. 24, pp. 40–49, 2014.

[6] R. Huang, H. Tawfik, and A. K. Nagar, "A novel hybrid artificial immune inspired approach for online break-in fraud detection," *Procedia Computer Science*, vol. 1, no. 1, pp. 2733–2742, 2010.

[7] A. Kundu, S. Panigrahi, S. Sural, and A. K. Majumdar, "Blast-ssaha hybridization for credit card fraud detection," *IEEE transactions on dependable and Secure Computing*, vol. 6, no. 4, pp. 309–315, 2009.

[8] M. Krivko, "A hybrid model for plastic card fraud detection systems," *Expert Systems with Applications*, vol. 37, no. 8, pp. 6070–6076, 2010.

[9] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar, "Credit card fraud detection using hidden markov model," *IEEE Transactions on dependable and secure computing*, vol. 5, no. 1, pp. 37–48, 2008.

[10] J. K.-F. Pun, "Improving credit card fraud detection using a meta-learning strategy," Ph.D. dissertation, 2011.

[11] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[12] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[13] R. Saia and S. Carta, "Evaluating credit card transactions in the frequency domain for a proactive fraud detection approach." in *SECRYPT*, 2017, pp. 335–342.

[14] I. Mani and I. Zhang, "knn approach to unbalanced data distributions: a case study involving information extraction," in Proceedings of workshop on learning from imbalanced datasets, vol. 126, 2003

[15] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). IEEE, 2008, pp. 1322–1328.

[16] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in International conference on intelligent computing. Springer, 2005, pp. 878–887.

[17] H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline over-sampling for imbalanced data classification," in Proceedings: Fifth International Workshop on Computational Intelligence & Applications, vol. 2009, no. 1. IEEE SMC Hiroshima Chapter, 2009, pp. 24–29.