iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VIII **Month of publication:** August 2022

DOI: https://doi.org/10.22214/ijraset.2022.46421

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VIII Aug 2022- Available at www.ijraset.com

Juxtaposing Sampling Techniques for Credit Card Fraud Detection

Aryan Ringshia[1], Neil Desai[2], Prachi Tawde[3]
[1, 2, 3]*Department of Information Technology, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India*

Abstract: ***Dealing with the class imbalance issue in the data is a big problem when creating fraud detection systems since valid transactions exceed fraudulent transactions by a large degree. Fraudulent transactions generally make up less than 1% of all transactions. This is a crucial field of research because it can be challenging to discern between a positive instance (a fraudulent case), and it gets tougher when more data are collected and a smaller percentage of such cases are represented. Eight distinct sampling techniques and two classifiers were used in this investigation, and the results of each methodology are reported. The results of this study point to favourable outcomes for sampling methods based on SMOTE. The best F1 score score obtained was with SMOTE sampling strategy on Random Forest classifier at 0.867.***
Keywords: Data Imbalance, Sampling Techniques, Classification, Fraud Detection, Supervised Learning

I. INTRODUCTION

Any credit card firm places an emphasis on spotting fraudulent transactions. In order to prevent customers from being charged for products they did not buy, credit card firms must be able to identify fraudulent credit card transactions. Credit card fraud often involves an unlawful transaction made by someone who is not authorised to handle that particular account's business. It may also be categorised when someone uses a card to make a purchase without the cardholder's or card issuer's express consent.

An example of a classification problem where the distribution of examples among the recognised classes is biased or unbalanced is an imbalanced classification problem. There is an uneven distribution of the classes. [1] Many real-world classification problems have an imbalanced class distribution specially where anomaly detection is crucial, such as:

1) Identification of rare diseases like cancer; tumours etc,
2) Electricity theft & pilferage
3) Fraudulent transactions in banks
4) Identify customer churn rate (that is, what fraction of customers continue using a service)
5) Natural Disasters like Earthquakes
6) Spam emails, etc.

Predictive modelling is challenged by imbalanced classification issues because the majority of machine learning methods for classification were built on the premise that there should be an equal number of samples in each class. [2]

An unbalanced classification predictive modelling problem may have several root causes for the class distribution imbalance. [3] It's likely that the manner the examples were gathered or sampled from the problem area contributed to the imbalance in the examples across the classes. This may involve biases imposed and mistakes committed during the data collection process.

a) Biased Sampling.
b) Measurement Errors

By using better sampling techniques and/or correcting the measurement mistake, the imbalance that is being generated by a sample bias or measurement error can be resolved. Another possible cause could be that imbalance might be a property of the problem domain.[4] For example, the natural occurrence or presence of one class may dominate other classes. (Anomaly detection) [4]

Imbalanced classifications pose a challenge for predictive modelling as most of the machine learning algorithms used for classification were designed around the assumption of an equal number of examples for each class. Decision Tree and Logistic Regression are two examples of common classifier algorithms that are biased in favour of classes with more occurrences. They frequently only forecast the bulk of the class data. The characteristics of the minority class are frequently dismissed as noise. As a result, the minority class is more likely to be incorrectly classified than the dominant class. As a result, models perform poorly in terms of prediction, particularly for the minority class. This is a problem because typically, the minority class is more important and therefore the problem is more sensitive to classification errors for the minority class than the majority class. [5]

II. LITERATURE REVIEW

There are several techniques to handle the imbalance in a dataset.[6] Re-sampling, a data-level strategy, is one of the most applied and well-liked techniques. Prior to supplying the data as input to the machine learning algorithm, we concentrate on balancing the classes in the training data (data pre-processing). The main objective of balancing classes is to either increase the frequency of the minority class or decrease the frequency of the majority class. [7]

A. Under sampling

The majority class points in the data are resampled while undersampling to be equivalent to the minority class points. By employing undersampling, the original dataset is transformed into a new dataset. To more evenly distribute the classes, it eliminates examples from the training dataset that belong to the dominant class. The main drawback of undersampling is that a sizeable portion of the data, which contains some information, is not utilised.

1) *Random Under Sampling:* Random Undersampling aims to balance class distribution by randomly and uniformly eliminating majority class examples. This is carried out until the instances of the majority and minority classes are equal. By lowering the number of training data samples when the training data set is large, it can help with run time and storage issues. On the other hand, it may ignore data that might be valuable and crucial for creating rule classifiers.
2) *Near Miss method:* It is a collection of undersampling techniques that choose samples based on the separation between instances from the majority and minority classes. The method determines how far every member of the majority class is from every member of the minority class. The next step is to choose the k instances of the majority class that are closest to those in the minority class. The "nearest" approach will produce k*n instances of the majority class if there are n instances of the minority class. The majority class examples that are the three nearest minority class examples with the shortest average distance are chosen by NearMiss-1. In order to choose examples from the majority class, NearMiss-2 compares them to the three instances from the minority class that are closest to them on average. NearMiss-3 involves selecting a given number of majority class examples for each example in the minority class that are closest
3) *Edited Nearest Neighbor method (ENN):* Based on the notion of nearest neighbour (NN), the objective behind this strategy is to remove examples from the majority class that are close to or near the boundary of distinct classes in order to improve the classification accuracy of minority instances rather than majority instances. [8]. The ENN method works by finding the K-nearest neighbor of each observation first, then check whether the majority class from the observation's k-nearest neighbor is the same as the observation's class or not. The rule can be applied to each example in the majority class as part of an undersampling approach, allowing those that are incorrectly identified as belonging to the minority class to be eliminated and those that are correctly identified to be kept. Additionally, it is applied to every case in the minority class, where the closest neighbours from the majority class are removed from the examples that were incorrectly classified. The n_neighbors argument controls the number of neighbors to use in the editing rule, which defaults to three.

B. Over Sampling

1) *Random over Sampling:* A single instance may be chosen more than once because Random Oversampling involves selecting random instances from the minority class with replacement and augmenting the training data with multiple copies of this instance. Since random oversampling creates precise replicas of the minority class samples, it may increase the risk that overfitting will occur. The higher computing cost is another issue we need to be mindful of. When training our model, increasing the number of examples in the minority class (especially for a badly skewed data set) may raise the computation cost, which is undesirable because the model would see the same cases repeatedly. However, For Machine Learning algorithms affected by skewed distribution, such as artificial neural networks and SVMs, this is a highly effective technique.
2) *SMOTE [9]:* SMOTE is short for Synthetic Minority Oversampling Technique. This method is used to prevent the overfitting that results from adding exact clones of minority occurrences to the main dataset. The minority class's subset of data is used as an example before additional artificial instances that are comparable to it are produced. These are not copies or reproductions of minority class data that already exist. The initial dataset is then updated with these created instances. The classification models are trained using a sample from the fresh dataset. SMOTE selects examples in the feature space that are close to one another, draws a line between the examples, and then creates a new sample at a location along the line. First, a representative from the minority class is picked at random. Next, k nearest neighbours for that example are located (k is normally equal to 5). A synthetic example is made at a randomly chosen position in feature space between two instances and their randomly chosen neighbour. This procedure can be used to create as many synthetic examples for the minority class as are required. The strategy

works because it generates convincing new synthetic examples from the minority class that are substantially near in feature space to already existing examples from the minority class. Synthetic examples are constructed without taking into account the majority class, which could lead to misleading examples if there is a significant overlap between the classes.

3) *ADASYN [10*]: Short for Adaptive Synthetic Sampling Approach, a generalization of the SMOTE algorithm. This technique also aims to oversample the minority class by creating virtual instances of it. The difference in this case is that it takes into account the density distribution, which determines how many artificial instances are created for confusing samples. As a result, it is beneficial to alter the decision constraints in light of the challenging samples. The more general framework ADASYN determines the impurity of the neighbourhood for each minority observation by dividing the number of majority observations in the neighbourhood by k.

C. Hybrid Sampling [11]

1) *SMOTEENN:* SMOTE + ENN is another hybrid technique where a greater number of observations are removed from the sample space. Here, the nearest neighbours of each member of the majority class are calculated using ENN, another undersampling technique. The majority class instance is eliminated if the closest neighbours incorrectly classify that specific instance. This method can be used in conjunction with SMOTE's oversampled data to perform thorough data cleaning. Samples from both classes that were incorrectly classified by NN's are eliminated here. As a result, the class distinction is more distinct and succinct.
2) *Combining Both Random Sampling Techniques:* To enhance the bias toward these cases, a small amount of oversampling can be applied to the minority class, and a small amount of undersampling can be applied to the majority class to lessen the bias on that class. Compared to using just one or the other technique alone, this may lead to better overall performance.

III. METRICS FOR EVALUATION

Accuracy is no longer a valid metric in the context of unbalanced data sets since it does not account for the proportion of examples from various classes that are correctly categorised. As a result, it could result in incorrect inferences. Even with a somewhat skewed class distribution, accuracy can still be a helpful metric. Accuracy can stop being a trustworthy indicator of model performance when the class distributions are severely skewed. For categorization that is not balanced, confusion matrix accuracy is meaningless.

A. Recall

Recall gives us the response to a different query, "What fraction of all of the positive samples did model accurately predict?" We are now concerned in false negatives rather than false positives. These are the faults that our algorithm failed to detect and are frequently the most egregious ones (e.g., failing to diagnose something with cancer that actually has cancer, failing to discover malware when it is present, or failing to spot a defective item). In this case, the term "recall" also makes sense because we are examining the proportion of samples that the algorithm was able to identify.

B. Precision

What percentage of the positive predictions made by the model are accurate? can be answered using precision. The precision will be low if your algorithm properly predicts every member of the positive class while also producing a sizable number of false positives. Given that it is a gauge of how 'precise' our forecasts are, it makes sense why this is termed precision.

C. F1 score

The F1 score is a single-value statistic that utilises the harmonic mean to combine precision and recall (a fancy type of averaging). The parameter, which has a strictly positive value, is used to express how important recall is in comparison to precision. A bigger value places more weight on recall than precision, whereas a smaller value places less weight on recall. If the value is 1, equal weight is given to both precision and recall.

What does an excellent F1 score imply? It implies that both the recall and precision have high values, which is positive and what you would expect to observe after creating a successful classification model on an unbalanced dataset. Low values signify poor memory or precision, which may be cause for concern.

Good F1 scores typically perform worse than good accuracy (in many situations, an F1 score of 0.5 would be considered pretty good, such as predicting breast cancer from mammograms).

D. AUC score

AUC-ROC is the valued metric used for evaluating the performance in classification models. The AUC-ROC statistic definitely aids in determining and informing us about a model's capacity for classifying data. Higher AUC indicates a better model, according to the grading standards. The relationship and trade-off between sensitivity and specificity for each feasible cut-off for a test being conducted or a set of tests being run are typically shown graphically using AUC-ROC curves. The advantage of utilising the test for the underlying question is indicated by the area under the ROC curve. At various threshold levels, AUC-ROC curves are another performance metric for classification issues.

IV. DATASET

The dataset of credit card fraud detection is taken from Kaggle. The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. It contains only numerical input variables which are the result of a PCA transformation. The only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise. The total legit transactions are 284315 out of 284807, which is 99.83%. The fraud transactions are only 492 in the whole dataset (0.17%) making it a highly imbalanced dataset.

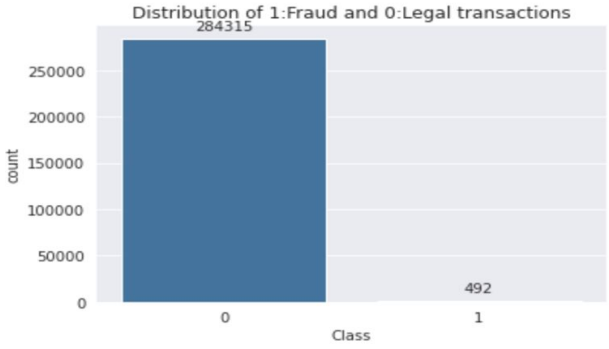


Fig. 1 Distribution of transactions in dataset.

V. RESULTS

TABLE I
RESULTS

Method	Model	F1 Score	AUC Score
Random Undersampling	Random Forest Classifier	0.187	0.788
Near miss	Random Forest Classifier	0.044	0.764
ENN	XG Boost Classifier	0.847	0.861
Random Oversampling	XG Boost Classifier	0.867	0.849
SMOTE	Random Forest Classifier	0.867	0.859
ADASYN	Random Forest Classifier	0.854	0.865
SMOTEENN	Random Forest Classifier	0.837	0.856
Hybrid Random	Random Forest Classifier	0.850	-

VI. CONCLUSIONS

Credit card fraud detection is categorised as a cost-sensitive topic because there is a cost involved in mistakenly identifying a legitimate transaction as fraudulent and a fraudulent transaction as real. Financial institutions do not pay any related administrative costs when fraud is absent or does not occur. The specific transaction value is lost if the scam is not discovered. False Positives, in which legitimate transactions are marked as fraudulent, have a cost. On the other hand, the cost of failing to spot a fraudulent transaction might be quite high.

A base model was implemented using an unsampled dataset, followed by the implementation of eight different sampling strategies. For each of the sampled datasets, two classifiers—Random Forest Classifier and XG Boost Classifier—were used. Each was assessed based on their F1 and AUC scores. SMOTE sampling strategy produced the best results, hence it can be regarded as a more effective sample strategy to use.

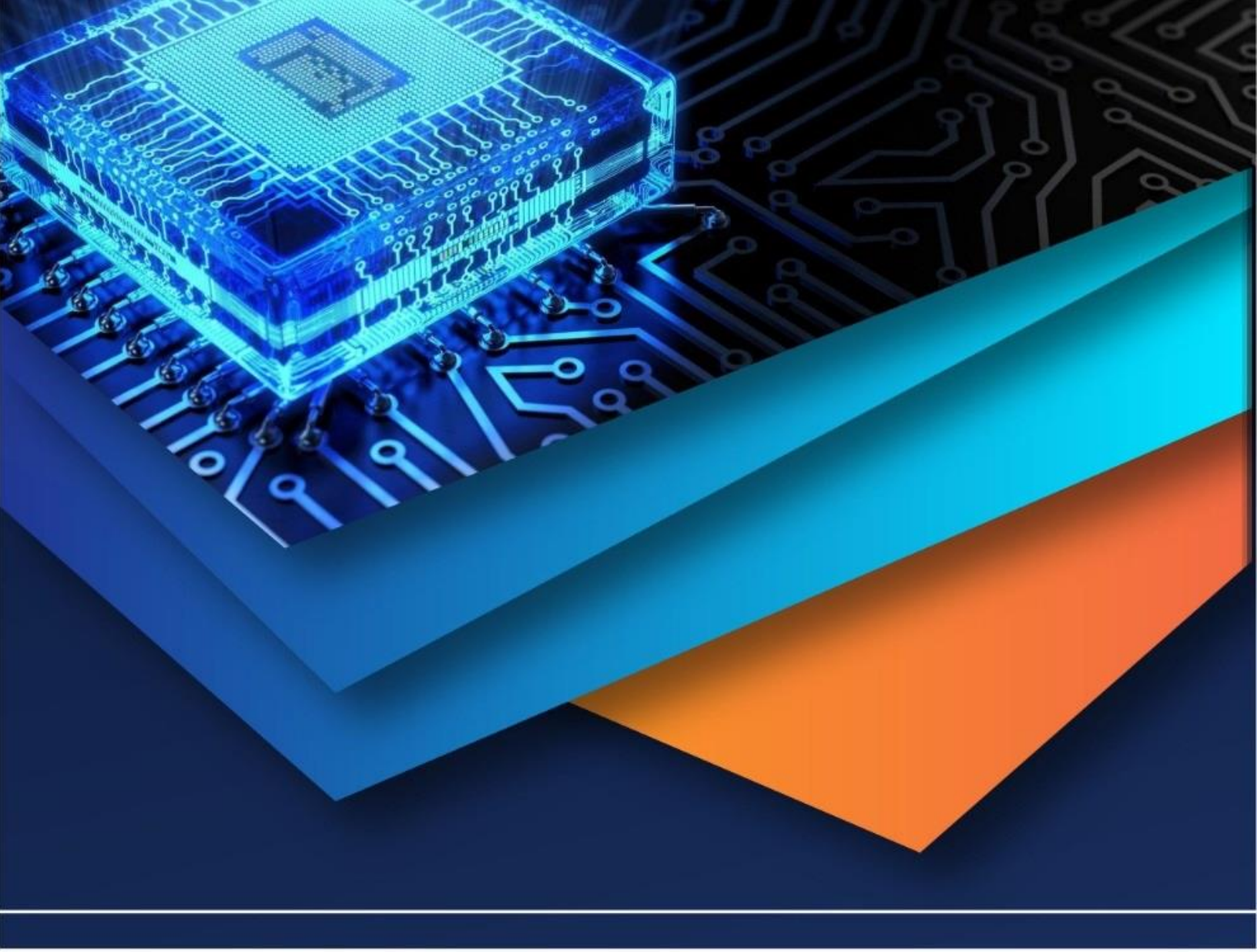
Despite the fact that the study's primary goal was to address the principal issues associated with anticipating fraudulent transactions, the study's time and resource constraints led to the selection of a small number of sample strategies. A number of other sample techniques might be taken into account as a direction for future study to enhance the performance of the classifier. Unsupervised machine learning was not included in the purview of this study, but it is still a promising area that must be explored. The application of semi-supervised or unsupervised learning approaches, such as one-SVM, k-means clustering, and isolation forests, may further enhance this work. Finding the ideal thresholds for identifying the cut-off points to maximise the recall score and striking the proper balance between precision and recall are two other areas where research can be expanded to produce possibly beneficial outcomes.

VII. ACKNOWLEDGMENT

We would like to express our appreciation for Professor Prachi Tawde, our research mentor, for her patient supervision, ardent support, and constructive criticisms of this research study.

REFERENCES

[1] V. Jayaswal, (2020) Dealing with imbalanced dataset on Towards Data Science. [Online]. Available: https://towardsdatascience.com/dealing-with-imbalanced-dataset-642a5f6ee297
[2] M. Tripathi (2022) Understanding Imbalanced Datasets and techniques for handling them on DataScience Foundation. [Online]. Available: https://datascience.foundation/sciencewhitepaper/understanding-imbalanced-datasets-and-techniques-for-handling-them
[3] J. Brownlee (2020) A Gentle Introduction to Imbalanced Classification on Machine Learning Mastery. [Online]. Available: https://machinelearningmastery.com/what-is-imbalanced-classification/
[4] W. Badr (2019) Having an Imbalanced Dataset? Here Is How You Can Fix It on Towards Data Science. [Online]. Available: https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb
[5] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," *GESTS International Transactions on Computer Science and Engineering*, vol. 30, pp. 25-36, Nov. 2005.
[6] Taherdoost, Hamed, "Sampling Methods in Research Methodology; How to Choose a Sampling Technique for Research," *International Journal of Academic Research in Management*, vol. 5, no. 2, pp. 18-27, Apr. 2016.
[7] Y. Sun, A. Wong, and M. Kamel, "Classification of imbalanced data: a review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 04, pp. 687-719, Nov. 2011.
[8] D. L. Wilson, "Asymptotic Properties of Nearest Neighbor Rules Using Edited Data," in IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-2, no. 3, pp. 408-421, July 1972.
[9] Han, H., Wang, WY., Mao, BH. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: Huang, DS., Zhang, XP., Huang, GB. (eds) Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science, vol 3644. Springer, Berlin, Heidelberg.
[10] Haibo He, Yang Bai, E. A. Garcia and Shutao Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 2008, pp. 1322-1328, doi: 10.1109/IJCNN.2008.4633969.
[11] C. Seiffert, T. M. Khoshgoftaar and J. Van Hulse, "Hybrid sampling for imbalanced data," 2008 IEEE International Conference on Information Reuse and Integration, 2008, pp. 202-207, doi: 10.1109/IRI.2008.4583030.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY