ADASYN for Imbalanced Learning in Big Data

Lim Yi Yong
School of Computer Science
University of Nottingham
Nottingham, United Kingdom
hcyyl4

Khor Yong Teng
School of Computer Science
University of Nottingham
Nottingham, United Kingdom
hfyyk4

Lau Yu Xuan
School of Computer Science
University of Nottingham
Nottingham, United Kingdom
hcyyl3

Ang Jia Hau
School of Computer Science
University of Nottingham
Nottingham, United Kingdom
hcyja1

***Abstract* - ADASYN is well known for its effectiveness in solving highly imbalanced datasets. It oversamples the minority classes to eliminate the bias of machine learning models towards the majority classes. However, with large and highly imbalanced datasets, using ADASYN potentially causes the dataset to double in size, significantly increasing the computational effort required for data processing and machine learning. In this work, we present a big data approach for imbalanced learning implemented under PySpark. The aim of this project is to investigate the feasibility of our big data approach, utilizing ADASYN oversampling technique. From our conducted research, increasing in number of partitions yields greater efficiency during computation. SVM with 11 partitions achieved the best F1-Score which is 0.0623 with an accuracy of 74.23% under 14 minutes 39 seconds. As the number of partitions increases from 11 to 19, the time taken reduces to 5 minutes 39 seconds with an F1-Score of 0.0589 and an accuracy of 75.52%. Therefore, showcasing the reliability and efficiency of our proposed approach.**

I. Introduction

Fraud detection is a set of preventive measures undertaken to protect a customer's assets. It is applied to many industries such as banking, insurance and healthcare sectors. A fraud can be committed in different ways. A common type of fraud in the banking industry is known as transaction fraud. This happens when someone illegally gains access to a victim's account without the real account holder's knowledge to generate an unauthorised transaction. To differentiate a legitimate transaction from a fraudulent transaction, it is necessary to verify the information obtained from a variety of sources. For example, user identity, transaction details and merchant's sales record. This helps us to obtain a useful insight and recognise the descriptive properties of a transaction. However, the amount of data collected will be increasing exponentially as the time goes on. Especially the transactions nowadays are moving towards real-time. It increases the chances of fraudulent transactions being undetected if the fraud detection isn't capable of processing a huge amount and variety of data in a short amount of time. Based on Fraud- The Facts 2021[1], a failure in fraud detection in industries has cost a total loss of 1.26 billion in the UK in 2021. As a result of the factors mentioned previously, fraud detection is considered a big data problem because it satisfies the 5 V's characteristics of big data which are veracity, variety, volume, velocity and value [2]. Therefore, a big data approach should be taken to tackle the problem.

The number of legitimate transactions will always be higher than the number of fraudulent transactions. Therefore, using the original distribution of two classes leads to an imbalanced dataset. It causes the machine learning model to perform poorly and have a bias towards predicting the majority classes in all cases. Nowadays, there are two commonly used techniques to handle the problem of class imbalance which are oversampling and undersampling. In [3] and [4], they show the performance of oversampling technique is overall better than the sampling technique. Although it generates more data, it behaves in a robust manner in a noisy environment and allows the machine learning models to achieve a significant result. Among the oversampling techniques, Adaptive Synthetic (ADASYN) is often used to fix the imbalance dataset [5]. It is built on the methodology of an oversampling technique called Synthetic Minority Oversampling technique (SMOTE) [6]. ADASYN adaptively generates an arbitrary number of new synthetic minority examples to achieve a balance between the classes.

A. Related Works

ADASYN is an oversampling technique that was first proposed in [5], as a means to improve machine learning on imbalanced data sets, by reducing the bias introduced by the class imbalance and adaptively shifting the classification decision boundary toward difficult examples. ADASYN has been utilised in various works such as churn prediction [7], increasing F1-Score performance up by 80.396%, and detection of fake accounts [8] and often compared to the SMOTE oversampling technique as seen from these works.

As stated in [9] imbalanced Big Data classification poses its own set of challenges especially in terms of accuracy and efficiency. The more relevant of which is that different levels of partitioning must be considered to maintain the robustness of modelling when seeking for higher scalability and predictive performance. Investigations in imbalanced Big Data classification have been done using different sampling techniques as well in [10] which includes experimentation with ADASYN.

B. Aims and Objectives

The aim of this project is to utilise ADASYN for the challenge of imbalanced class samples within a given dataset for transaction fraud detection, which deals with minority interests and rare instances. The volume of transactions

highlights the requirement for big data techniques when building the methodology.

To achieve this aim, the following key objectives will need to be met:

1. ADASYN is applied efficiently for oversampling.
2. Efficiently carry out big data processing using a local approach
3. Naïve bayes, Gradient Boosting Classifier and SVM are then experimented and compared for the binary classification which identifies fraudulent and non-fraudulent transactions.

II. METHODOLOGY

Dataset pre-processing is first carried out to ensure the subsequent analysis of information performed is relevant. This includes the combination of two distinct datasets, followed by a series of operations to produce a final output dataset. A local approach through divide and conquer, whereby an ensemble of models created from each partition of the dataset is utilized through an ensemble strategy of majority voting and aggregation is then carried out.

A. Pre-processing

The pre-processing stage consists of a total of 8 stages. Initially, the identity and transaction datasets are treated as initial inputs which are then subsequently combined to produce the final input dataset. Columns with more than 90% of null values are then dropped, followed by the mapping of numerical values from string values. Remaining null values within each column are then filled using an Imputation estimator which uses the *mean* values of columns. To prepare the dataset for machine learning tasks, the datatypes of all columns are converted to *float* types except isFraud (target label). *Vector Assembler* is then used to merge columns to a vector column, allowing for the utilization of PCA (Principal Component Analysis) for feature reduction, which then produces the final output dataset. Figure 1 illustrates the pre-processing flowchart.

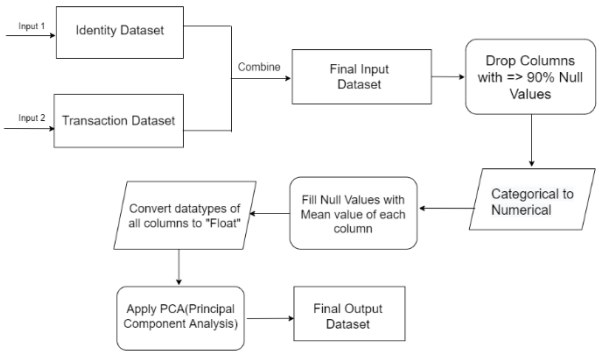

Figure 1 : Pre-processing flowchart

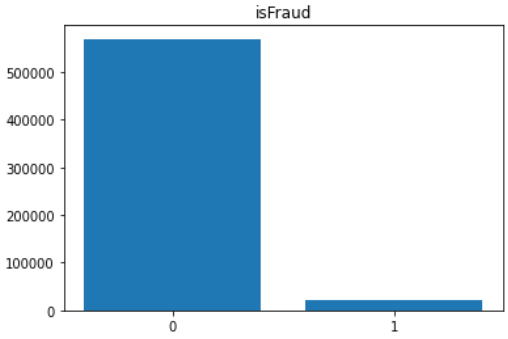

Figure 2 - An overview of imbalanced dataset

B. Partitioning & Training

After the pre-processing stage, ADASYN is not immediately implemented as it would require a lot of computational effort to generate sufficient synthetic data for a large and severely imbalanced data set. Instead, the final dataset is split into a "Fraud" dataset and a "Not Fraud" dataset, such that the minority class is separated from the majority. The "Not Fraud" dataset being the majority is then partitioned to γ number of partitions. The minority class is then broadcasted to the γ number of partitions, followed by ADASYN as an oversampling method for synthetic data generation to strategically deal with the class imbalance. It ensures that data from the minority class to be present in every partition. Lastly, γ number of models is trained from the balanced dataset, in which we compared Gradient Boosting Tree, Naïve bayes and SVM (Support Vector Machine). Figure 2 illustrates the partitioning and model training stage.

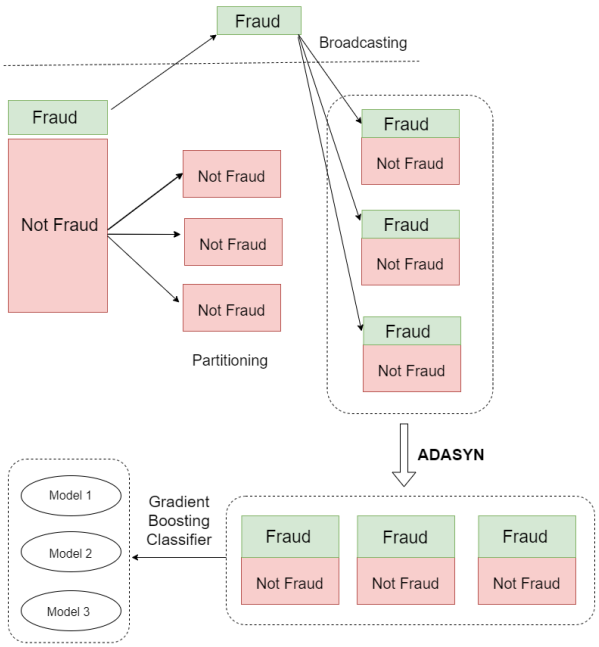

Figure 3 : Partitioning and Training stage

C. ADASYN

ADASYN (Adaptive Synthetic) algorithm is an adaptive oversampling approach used to generate minority class samples within an imbalanced dataset which are harder to learn. Not only does this reduce the bias within the original

imbalanced dataset, ADASYN is also able to shift the focus of decision boundaries towards samples which are more challenging to learn.

An input dataset D_{tr} containing minority class sample m_i and majority class sample m_a will first have its degree of class imbalance measured through the calculation of the ratio of between the minority and majority class, given in Equation 1.

$$d = \frac{m_i}{m_a} \ldots\ldots (1)$$

A threshold value given by d_t is then taken and measured against the calculated degree of class imbalance, d. If the degree of imbalance is not lower than the threshold, then the dataset has attained a degree of balance, and there are no further operations to be considered. Otherwise, the oversampling process will be executed.

The difference of the total instance between the majority and minority class is calculated. The difference is then multiplied by a value, β, to control the amount of synthetic data to be generated, given in Equation 2.

$$G = (m_a - m_i) \times \beta \ldots\ldots (2)$$

We then find the k – nearest neighbors of each entry from the minority class. From the i-th neighborhood, the number of the majority classes in the neighborhood is calculated and stored as Δi. By dividing it with k, we can determine the ratio of the majority class in the i-th neighborhood, r_i, as shown in Equation 3.

$$r_i = \frac{\Delta_i}{K}, i = 1, \ldots, m_s \ldots\ldots (3)$$

The sum of r_i, Σr_i, is calculated. Each r_i are then normalized by dividing with Σr_i and stored as r_i^{λ}, such that $\sum r_i^{\lambda} = 1$, as described in Equation 4.

$$r_i^{\lambda} = \frac{r_i}{\Sigma r_i} \ldots\ldots (4)$$

By normalizing, we can now accurately determine the amount of synthetic data to be generated within each neighborhood. This can be calculated by the product of r_i^{λ} and G. The product is then stored as g_i, as seen in Equation 5.

$$g_i = r_i^{\lambda} \times G \ldots\ldots (5)$$

For a vectorized sample from the minority class, x_i, we find another vectorized sample from the minority class, x_{zi} in its neighborhood. The distance between x_i and the synthetic data to be generated can be calculated by the product of a random value between 0 and 1, and the vector difference between x_{zi} and x_i. Add the distance calculated to x_i, and the vectorized synthetic minority example, s_i, is generated. A simple representation is shown in Equation 6. The synthetic minority samples are then merged with the original dataset.

$$s_i = x_i + (x_{zi} - x_i) \times \lambda,\ \lambda \in [0,1] \ldots\ldots (6)$$

Figure 3 illustrates the flowchart for the ADASYN algorithm.

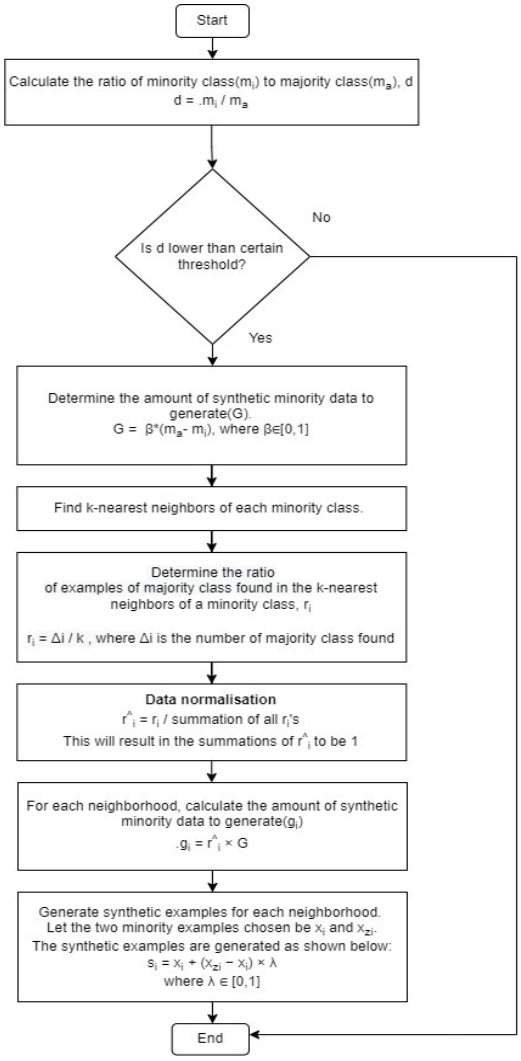

Figure 4 : ADASYN Flowchart

D. Testing

After pre-processing, the dataset is partitioned directly without the separation of minority and majority classes. This results in the potential of certain partitions not consisting of the minority class. The γ number of models are then applied directly on each partition to produce γ number of predictions for each test data. An ensemble strategy through the aggregation of results from the models is then carried out through majority voting, which produces the final prediction set. Figure 4 illustrates the testing process.

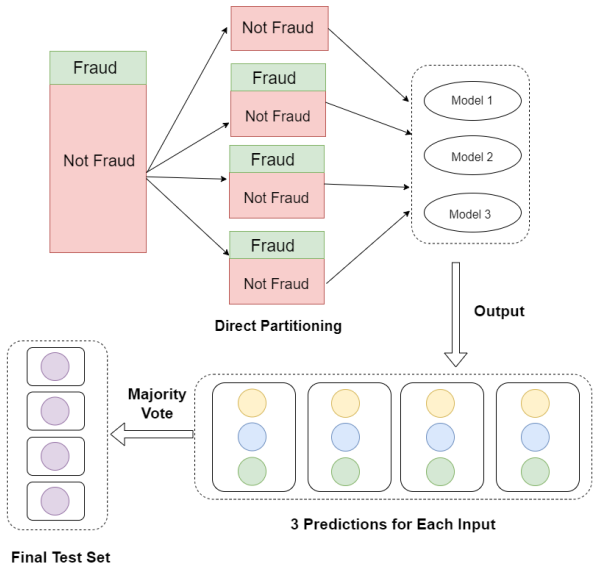

Figure 5 : Testing methodology

III. Experimental Set Up

We select the fraud detection datasets provided by [11] as our test subject. It is a highly imbalanced dataset with 569877 instances of "not fraud" class, 20663 instances of "fraud" class and 434 features. Three machine learning models (Gradient Boosting Classifier, Naïve Bayes and SVM) are selected to evaluate the effectiveness of ADASYN. We have taken a local approach which mainly based on the number of partitions. The experiment take place on Databricks using a cluster of 88GB and 24 cores.

IV. Result and Evaluation

Evaluation metrics such as accuracy, precision, recall and F1-score were calculated from the predictions obtained using different classifiers and number of partitions. Results recorded are presented in the table below, along with the time taken to complete the partitioning, broadcasting, ADASYN, and modelling processes.

Number of partitions	11	15	19
Gradient Boosting Classifier	Accuracy : 69.11% F1 : 0.0609 Precision : 0.0341 Recall : 0.2872 Time taken : 6min27sec	Accuracy : 66.76% F1 : 0.0610 Precision : 0.0340 Recall : 0.3001 Time taken : 4min36sec	Accuracy : 70.57% F1 : 0.0589 Precision : 0.0332 Recall : 0.2624 Time taken : 3min36sec
Naïve Bayes	Accuracy : 86.87% F1 : 0.0492 Precision : 0.0329 Recall : 0.0970 Time taken : 4min22sec	Accuracy : 87.31% F1 : 0.0532 Precision : 0.0359 Recall : 0.1032 Time taken : 3min5sec	Accuracy : 88.44% F1 : 0.0565 Precision : 0.0392 Recall : 0.1009 Time taken : 2min31sec
SVM	Accuracy : 74.23% F1 : 0.0623 Precision : 0.0357 Recall : 0.2440 Time taken : 14min39sec	Accuracy : 72.50% F1 : 0.0613 Precision : 0.0348 Recall : 0.2588 Time taken : 9min6sec	Accuracy : 75.52% F1 : 0.0589 Precision : 0.0341 Recall : 0.2192 Time taken : 5min39sec

Upon inspecting the results, it can be observed that the time taken for processing decreases significantly with the increase of partitions used. SVM takes the longest to run among all modelling approaches, followed by Gradient Boosting Classifier (GBC) and then Naïve Bayes. Being an imbalanced classification problem, F1-score, precision, and recall serves as a better evaluation metric for model performance instead of accuracy. The performance of the classifiers are poor as observed, only achieving a maximum of F1 = 0.0623 with SVM on 11 partitions. When increasing the number of partitions, the F1-score of GBC and SVM decreases but surprisingly increases for Naïve bayes, likely caused by the increase in precision observed. Accuracy observed is rather varied as it drops between 11 partitions to 15 partitions for GBC. Overall, SVM seems to be the best performing classifier, followed by GBC then Naïve bayes. In summary, the increase in partitions slightly decreases the performance of classifiers except in the case of Naïve bayes (Obtained from F1-score) but significantly reduces the time taken for computation. The increase in efficiency is likely due to the "divide and conquer approach" utilized and the decrease in dataset size per partition due to the division as well as the amount of minority class needed to be up sampled through ADASYN.

V. Conclusion

We have successfully proved the feasibility of our big data approach with regards to the computational efficiency achieved. It can process a large-scale dataset within an acceptable time. The performance of the machine learning models, however, leaves much to be desired in the case of this dataset used. As future work, we would like to further investigate our proposed big data approach on different datasets to further understand its feasibility in terms of predictive performance that the models are able to achieve. Other sampling methods such as oversampling using SMOTE or under sampling techniques could be experimented as well. Besides that, the utilization of global approach may also boost the data processing in big data problems.

VI. References

[1] Katy Worobec , "Fraud-The Facts 2021" , UK FINANCE, vol/version, page 16. , 2021

[2] Ishwarappa, Anuradha."A brief introduction on big data 5Vs characteristics and Hadoop Technology". Procedia Computer Science. Volume 48. Pages 320-321. 2015

[3] Roweida Mohammed, Jumanah Rawashdeh, Malak Abdullah."Machine Learning with Oversampling and Undersampling Techniques : Overview Study and Experimental Results". IEEE. page 5. 2020.

[4] Prabhjot Kaur, Anjana Gosain."Comparing he Behavior of Oversampling and Undersampling Approach of Class Imbalance Learning by Combining Class Imbalance

Problem with Noise". Springer Nature Singapore Pte Ltd. page 29. 2018.

[5] Haibo He, Yang Bai, Edwardo A.Garcia, Shutao Li."ADASYN : Adaptive synthetic sampling approach for imbalance learning". IEEE Xplore. page 1323. 2008

[6] Nitesh V. Chawla, Kevin W. Bowyers, Lawrence O. Hall, W. Philips Kegelmeyer. "SMOTE : Synthetic Minority Oversampling technique". Journal of Artificial Intelligence Research 16. pages 328-331. 2002.

[7] Annisa Aditsania, Adiwijaya, Aldo Lionel Saonard. "Handling imbalanced data in churn prediction using ADASYN and backpropagation algorithm". IEEE explore. page 533-536, 2017

[8] Aleksey G. Marakhtanov; Evgeny O. Parenchenkov; Nikolai V. Smirnov . "Detection of Fictitious Accounts Registration".IEEE. page 226-230. 2021

[9] Alberto Fernández, Sara del Río, Nitesh V. Chawla & Francisco Herrera. "An insight into imbalanced Big Data classification: outcomes and challenges". Complex Intell. Syst. page 105 – 120. 2017.

[10] Tawfiq Hasanin, Taghi M. Khoshgoftaar, Joffrey L. Leevy & Richard A. Bauder. "Severely imbalanced Big Data challenges: investigating data sampling approaches". Hasanin et al. J Big Data. page 8 – 20. 2019

[11] https://www.kaggle.com/c/ieee-fraud-detection (last accessed May 2022)