# A New GAN-based data augmentation method for Handling Class Imbalance in Credit Card Fraud detection

Emilija Strelcenia
Department of Creative Technology
Bournemouth University
Bournemouth, United Kingdom
e-mail: strelceniae@bournemouth.ac.uk

Simant Prakoonwit
Department of Creative Technology
Bournemouth University
Bournemouth, United Kingdom
e-mail: sprakoonwit@bournemouth.ac.uk

***Abstract*-- One of the most common cybercrimes that people encounter is credit card fraud. Systems for identifying fraudulent transactions that are based on intelligent machine learning are particularly successful in real-world situations. Nevertheless, when creating these systems, machine learning algorithms face the issue of imbalanced data or an unbalanced distribution of classes. Because of this, balancing the dataset becomes a crucial sub-task. A review of cutting-edge methods highlights the necessity for a thorough assessment of class imbalance management techniques in order to create a smart and effective system to identify fraudulent transactions. The goal of the current study is to compare several strategies for dealing with class imbalance. Therefore, the present study compares the performance of our novel K-CGAN method with SMOTE, B-SMOTE, and ADASYN in terms of Recall, F1-score, Accuracy, and Precision. The result shows that novel K-CGANs generated high quality test dataset and performs better as compared to other resampling techniques.**

***Keywords— Financial activities; online payments; dataset; imbalanced heavily; classification techniques; credit card; fraud transactions; Sampling methods; Synthetic Minority oversampling; Resampling methods; Random over sampling and Random under sampling***

## I. INTRODUCTION

According to Statista research (2022), the global fraud detection business is valued at approximately $19.5 billion [1]. Forecasts indicate a rise of $63 billion by 2023 from the $19.5 billion in 2017. Detection of fraud on streams of payments is one of the economy's most important concerns. Further, supervised learning is a popular classification technique within the growing role of machine learning in decision-making across many domains. Learning algorithms and forecasting are the two distinct phases of supervised binary classification. In the training phase, a classifier is formed using a powerful classification technique on the available training data; in the predictions phase, the learned classifier is then employed to estimate the unknown data [2]. Various techniques, such as the multi-layer perceptron [3], the decision tree algorithm [4], and the support vector machine [5], have been developed to solve the classification issue. The fields of pattern recognition, fraud detection, and intrusion detection also make extensive use of these techniques. However, current algorithms are developed with equality between classes in mind. The methods' optimization aim is to optimise the classification accuracy of all samples.

The data utilized in real-world applications is typically unbalanced. For instance, software defect detection, where normal samples predominate and fault samples are relatively rare [6], network intrusion detection, where normal traffic data predominates over attack traffic data [7], and fraud detection, where abnormal data predominates over normal data [8]. In the presence of skewed data, the algorithm's final identification result is more likely to support the majority class and disregard the minority class because it adds lesser to the total mistake [9]. It is possible for a classification algorithm to incorrectly label all data as on the majority class when there are just a small number of samples from the minority class. Minority-class samples are more valuable and should be prioritized in data extraction projects including fraud detection, intrusion detection, and defect detection among others [10].

When compared to the issue of classifying evenly distributed data, classifying unbalanced data is more challenging and complex [11]. It has become a tremendous challenge to increase overall recognition rates while also boosting those samples from under-represented groups. To address the issue of skewed data classification, numerous researchers have poured time and energy into studying the problem and proposing solutions in the form of algorithms. Data-level, algorithm-level, and ensemble learning are the three broad categories into which these techniques can be categorized [12]. Data resampling (under-sampling or oversampling) is a common data pre-processing technique that brings a dataset into statistical equilibrium. Algorithm-level strategies, in contrast to data-level pre-processing methods, typically involve designing new algorithms or improving existing algorithms (for instance; cost-sensitive techniques) to address the issue of imbalanced data classification [13]. Since Wasserstein-GAN has proven adept at fitting existing data, it has also been extensively studied for use in creating new data sets [14]. Class differences are just one cause of model learning difficulties, but studies of such

issues have shown that this is not always the case. To get the best overall classification result, a classifier will lean toward favouring the minority class when there are few examples of that class in the class overlapping area because of insufficient training [13]. As a result, class overlap greatly complicates training on skewed data [15]. The present study employs K-CGAN with different classification techniques including XGBoost, Random Forest, Nearest neighbour, MLP, and Logistic regression to generate synthetic data and detect frauds in credit card transactions. Furthermore, the current study compares and depicts the performance results of K-CGAN with SMOTE, B-SMOTE, and ADASYN.

## II. LITERATURE REVIEW

This section primarily provides a review and summary of the three most frequent approaches to addressing unbalanced data classification: the data-level approach, the algorithm-level approach, and the ensemble learning approach. Figure 1 shows major methods that handle class imbalance issues.

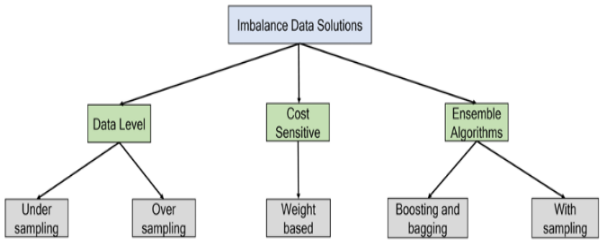


Fig. 1. Key methods to address class imbalance Source: [16]

### A. Data-level approach

The goal of a data-level technique is to resample the data until the negative and positive samples are about equal before classifying the data. Since the data-level processing approach is not dependent on the classification model, it is frequently employed to address the issue of unbalanced data sets that need to be categorized. Oversampling, under-sampling, and hybrid sampling are the primary categories into which the individual data-level approaches fall. The goal of oversampling is to correct this difference by increasing the size of positive samples while leaving negative samples alone. Ding et al. [9] argued that oversampling of positive samples may be broken down into two distinct categories: local information-based and global information-based. SMOTE oversampling is the most common technique used in local information-based oversampling [9]. By performing random linear interpolation between nearby samples, the SMOTE technique creates additional positive samples [17]. The SMOTE algorithm has inspired the development of many similar but more advanced approaches, including ADASYN, SMOTE-ENN, LORAS, and many more [18]. Although samples close to the decision boundary are crucial for classification, a lot of research has been done on how to reliably produce them [19]. The global oversampling technique generates new data by taking into account the variance, mean value, and probability distribution of positive samples, as opposed to generating new samples only based on local information [20]. The combined probability distribution of data characteristics and Gibbs sampling was introduced by Das et al. [21] as a method for creating new minority samples.

Since GAN has a high degree of accuracy when it comes to fitting data, it is often used for synthesis and the resolution of unbalanced learning issues [22]. Consider the sequence generative adversarial network-based credit default collection and analysis technique presented by Fan et al. [22] for the production of discrete data. In order to create credit default swap transaction data that is diverse and useful, this technique incorporates a reinforced learning approach into the original GAN network [23]. The reduction of negative samples (undersampling) helps even out the distribution of classes while preserving the integrity of the data (positive samples) [24]. Fundamental principle of undersampling is to eliminate samples that have negligible influence on the total data distribution, to maintain a balance between the positive and negative data [15]. Xie et al. [25] introduce density and distance as measures of sample significance, build a sampling sequence based on this importance, and then choose the most representative negative samples from this series. Further, data preparation can be improved via hybrid sampling, which combines oversampling and undersampling algorithms [26]. Class decomposition, as suggested by Elyan et al. [27] is one approach to optimizing classification accuracy when dealing with unbalanced data.

In order to deal with the problem of binary unbalanced data classification, Yang et al. [28] suggested a hybrid classifier ensemble architecture (HCE). Adaptive two-stage under sampling (ATUP) and metric-based data space transformation (MDST) are the core components of the methodology. For a well-rounded dataset, they employ MDST to locate the proper embedding space and ATUP to select representative samples [29]. Traditional oversampling approaches create fresh samples locally, leading to poor generalisation capacity and unable to deliver improved classification judgements, and are thus among the methods based on the data level that have the potential to improve accuracy. Using an undersampling technique typically involves throwing out relevant data, which might alter the original data's distribution. Because of this, standard GAN methods frequently experience model collapse and fail to account for the sparseness of positive class data in the class overlap region [9].

### B. Algorithm-level approach

The primary focus of the algorithm-level study is on enhancing an established classification method so that it can handle unbalanced data, with the ultimate goal of boosting minority class classification performance while still maintaining a high bar for overall accuracy [30]. The issue

of unbalanced data categorization may be addressed with cost-sensitive learning, which is currently one of the most popular algorithm-level techniques [9]. The cost-sensitive approach aids in improving the identification accuracy of positive samples by guiding the classifier to adjust the weight of incorrectly classified positive samples. For instance, Fu et al. [31] suggested a Cost-Sensitive Support Vector Machine (CSSVM), a cost-sensitive model that takes its cues from both support vector machines (SVM) and the asymmetric linear exponential (LINEX) loss function. By assigning a separate cost to each event, the model can perform instance-level sensitivity learning. To address the issue of unbalanced data categorization, the SVM classifier employs a cost-sensitive loss function to regulate the expense of misclassifying positive and negative samples. Two cost-sensitive KNN classifiers, Direct-CS-KNN and Distance-CS-KNN, were suggested by Zhang [32] to reduce the negative impact of incorrect labels.

The algorithmic level is both more intuitive and more productive than the data level [9]. As a result, it excels in the categorization of data within a given domain. Although it is possible to enhance algorithms, this is not always the best approach. It is clear from the concept of the cost-sensitive technique that providing the matching cost-sensitive matrix is crucial to the design of the algorithm. In a cost-sensitive matrix, the weight setting is often determined by domain specialists and is thus extremely domain-specific. As an added downside, budget-friendly learning methods developed for one area are famously hard to adapt for use in a different one.

### *C. Ensemble learning approach*

Ensemble learning is a technique to improve the learning effect in the end by training numerous weak classifiers, which lowers the potential deviation of a single classifier in handling unbalanced data [33]. The primary foundation for ensemble learning may be divided into two categories: bagging and boosting. The majority of the techniques that address the issue of classification of unbalanced data are built on the development of the Bagging and Boosting framework. [9] Boosting-type algorithms attempt to decrease the deviation of weak classifiers, develop specific approaches in the model process, give more weight to the samples with the highest error rate, and then integrate each basic model to generate the final judgment. The XGBoost [34], GBDT [35], and AdaBoost [36] techniques are considered boosting algorithms [37]. Bagging algorithms primarily employ several sub-sample sets after sampling to create various weak classifiers and combine the classifiers using the ensemble technique to provide the prediction outcomes [38]. Additionally, the training impact is compromised since each classifier often has an insufficient distribution of negative data [9].

## III. RESEARCH METHODS

In order to achieve a more balanced class distribution in the dataset, the current study recommends using the data-level technique here. The suggested technique makes use of both undersampling and oversampling approaches. Through the use of oversampling methods, the data from the minority class may be converted into the same number from the majority class. Undersampling, on the other hand, involves removing samples from the majority class until the dataset has a uniform distribution of classes. By using these sample methods, a new, up-to-date dataset may be generated with an equal number of negative and positive labels. The generated dataset is then utilized to train more accurate machine-learning classification algorithms. Figure 2 shows the diagram of the class imbalance issue.

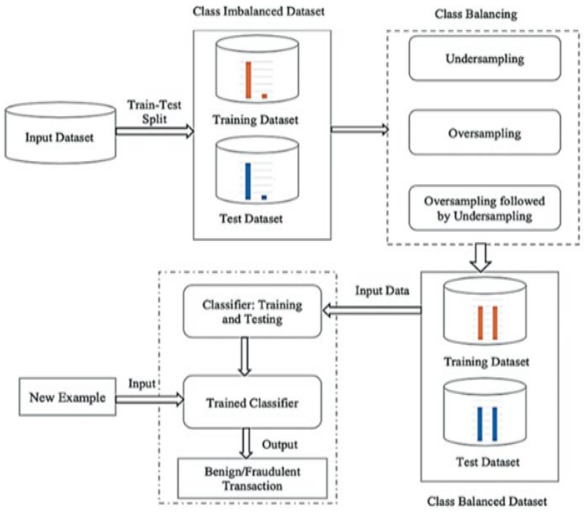


Fig. 2. The diagram of class imbalance issue. Source: [39]

First the present study split the class imbalance data set of credit card transactions into training and test data sets. Further, employs class balancing techniques (undersampling and oversampling) such as SMOTE, B-SMOTE, ADASYN, and novel K-CGAN with XGBoost, Random Forest, KNN, MLP, and Logistic Regression. Furthermore, it implies undersampling techniques followed by oversampling and acquires balanced datasets. Moreover, it trains the classifiers for training and testing datasets and finally identifies fraudulent transactions from synthetic data.

### *A. Experimental dataset*

There is a big problem for the research community, because financial institutes are not providing the required data for fraud detection. These financial institutes are bound

due to data security, sensitivity and due to privacy issues [9]. Therefore, it is quite difficult to obtain the required data. For this research we've used the publicly available Credit Card Transactions Fraud data and this training dataset consists of 23 informative fields that identify the credit card transaction time, merchant name and category, spending amount as well as personal particulars about the holders such as their names, genders, and ages. Additionally, there is an "is_fraud" column to flag any fraudulent transactions with a 1 or 0 respectively. As we have meticulously removed duplicates from the data set while eliminating missing data points altogether. Although the fraudulent transactions in our dataset constitute only 0.57%, this figure is significantly dwarfed by the 99.42% majority of non-fraudulent ones. Consequently, we must be sure to balance out data so that our analysis does not become contaminated with bias and misrepresentation.

### *B. Class Balancing techniques*

The present study uses novel K-CGAN, B-SMOTE, ADASYN, and SMOTE with XGBoost, Random Forest, Nearest neighbor, MLP, and Logistic regression classifiers to resolve the problem of class imbalance data and to detect frauds in credit card transactions.

## IV. EXPERIMENTAL DESIGN AND SETUP

Our proposed method K-CGAN which is based on Conditional GAN architecture (CGAN) with the custom loss function Kilberg divergence, hence the name K-CGAN. In order to achieve the best performance possible with our proposed method, many hyperparameters had to be adjusted. After extensive experimenting, we have found that the settings below work best and are shown in Table 1 and Table *2*. By utilizing the Weight Initialization (glorot_uniform) method, we were able to reduce the size of our neural network during training. We set the learning rate to 0.0001, hidden layer optimizer Relu, dropout ratio to 0.2 for both discriminator and generator hidden layers, bath size 64, number of epochs 500. The activation function was defined as Relu for generator and LeakyRelu for discriminator. Adam optimizer was used throughout this process. We also experimented with various Dropout values and discovered that a value of 0.2 achieved the best results. The creation of artificial data samples that adhere to the pattern "pg," which is statistically equivalent to the distribution of the real data, or "p data," and the simulation of scenarios in which a discriminator (D) network and a generator (G) network compete, are both made possible by generative adversarial networks. A discriminator network, on the other hand, is trained to tell the difference between real (derived from training data) and false (G-generated) samples. The K-CGAN employed for this work is based on conditional GAN architecture, where the cGAN training process is very similar to that of the GAN. A mini-batch of m training samples (xi, yi) mi=1 and m noise random samples zi, mi=1 is fed to produce the logistic cost function for the gradient. The generator attempts to produce data that is reasonably close to the training set to deceive the discriminator into classifying the dataset it generates as the training dataset.

TABLE I. GENERATOR NEURAL NETWORK HYPER PARAMETER SETTINGS

| Parameter | Value |
|---|---|
| Learning Rate | 0.0001 |
| Hidden Layer Optimizer | Relu |
| Output Optimizer | Adam |
| Loss Function | Trained Discriminator Loss+ **KL Divergence** |
| Dropout | 0.2 |
| Random Noise Vector | 50 |
| Kernel Initializer | glorot_uniform |

TABLE II. DISCRIMINATOR NEURAL NETWORK HYPER PARAMETER SETTINGS

| Parameter | Value |
|---|---|
| Learning Rate | 0.0001 |
| Hidden Layer Optimizer Leaky | LeakyRelu |
| Output Optimizer | Adam |
| Loss Function | Binary Cross Entropy |
| Dropout | 0.2 |

We can create fraud transactions by using the trained generator, and then include these fraudulent transactions in the real-time dataset. To calculate the K-CGAN generator and discriminator loss the following formulas were used.

### *A. Discriminator Loss*

If sample data x will be from real data, then the likelihood of the sample data x will be increased and if the sample data x will be from fake data, then the likelihood of the sample data x will be reduced. Following equation 1 is showing the discriminator loss:

$$Loss = -\frac{1}{\frac{output}{size}}\sum_{i=1}^{\frac{output}{size}} y_i \, . log\, y_i + (1 - y_i).\log(1 - y_i) \quad (1)$$

### *B. Generator loss*

The generator network is being used to generate fake data samples and these data samples are similar to the original data samples. Here, in this experience by a KL Divergence is being used in the equation. KL divergence is showing the difference of both distributions. Following equation 2 is

showing the generator loss calculation by adding the KL divergence.

$$Loss = -\frac{1}{\frac{output}{size}}\sum_{i=1}^{\frac{output}{size}} y_i \ .log\ y_i + (1 - y_i).\log(1 - y_i) + \sum p_i(x)\log\left(\frac{p_i(x)}{q_i(x)}\right) \quad (2)$$

The generator is being used to generate the sample data, while the optimizer is applied to reduce the kl-divergence to get extreme similar data.

### C. *Bordeline-SMOTE*

Even though Borderline-SMOTE can generate events that are technically in two separate categories, this is a very small fraction of all occurrences. B-SMOTE proposed by Chawla et al. [17]. In order to improve predictions, researchers used B-SMOTE during training to pinpoint the exact border between each class in most classification methods. B-SMOTE manufactures minority data in order to oversample the underrepresented group.

$$\mathrm{P} = \{\mathrm{p1}, \mathrm{p2}, \ldots\ldots, p_{pnum}\}, N = \{n1, n2, \ldots\ldots, n_{nnum}\} \quad (3)$$

Where n-num represents the total number of minority instances, and p-num represents the total number of majority cases.

### D. *ADASYN*

Both SMOTE and ADASYN have a common ancestor. However, ADASYN introduces a tiny random bias to the points after the samples are formed, making them less closely related to their parents. The variance of the synthetic data is increased although this is a small adjustment. To provide fake information, a synthetic adaptive algorithm is used to create minority data samples that have distributions that are typical of the underrepresented groups in order to address the data imbalance.

$$si = xi + (xzi - xi)\lambda \quad (4)$$

Minority cases xi and xzi in the same neighborhood as the innovative synthetic example si are generated using a random integer between 0 and 1.

### E. *SMOTE*

The technique is unequaled when it comes to learning from a wide variety of data sources. The equation for SMOTE may be written as:

$$xi = xi + \varsigma 1.(xi(nn) - xi) \quad (5)$$

The SMOTE technique will produce a new NT sample for a minority class if the total number of samples for that minority class with a training set is T. T=NT, the number of samples from a select few classes will be "thought" by the approach, which will also compel N=1. Let's say that, following the requirement that N must be a positive integer, N is presented as a negative integer. Consider Sample I from this minority class, which has the identifying vectors xi, i1..., T. All the t samples of this minority class contained the k neighbors of sample xi, which are identified as xi(near), near1..., k. These neighbors include the Euclidean distance, which was first determined in all the t samples. Use the sample xi(nn) that was arbitrarily selected from this k neighbor to generate a random number.

## V. CLASSIFIERS ANALYSIS

### A. *XGBoost*

This method enhances the original gradient-boosting approach. It enhances functionality overall by utilizing ensemble approaches. To address the problem of a non-uniform majority class, researchers adapt conventional classification algorithms using ensemble techniques.

$$(TOSi) = \frac{AUCi}{\sum_{i,j=1}^{k} |_{p(TOSi,TOSj)}|} \quad (6)$$

Where AUCi is the AUC efficacy of the i-th outlier identification approach and TOSi and TOSj denote the Pearson correlation coefficient between a pair of TOS.

### B. *Random Forest*

The Random Forest method for supervised machine learning may be used to solve classification and regression problems. During the training phase, it builds many decision trees and employs a majority vote to determine the conclusion in order to improve accuracy and produce more dependable forecasts.

$$IG(Np, a) = Gini\ (Np) - \sum_{i-1}^{c} \frac{|_{Ni}|}{|_{Np}|} Gini\ (Ni)$$

$$Gini\ (Np) = 1 - \sum_{j-1}^{m} p2j \quad (7)$$

where m stands for the number of different labels of data at node Np, and pj is the proportion of the number of data with the jth label over the total number of data at node Np. Np stands for the quantity of data at node Np, and |Ni| speaks for the amount of data at node Ni, 0Ic.

### C. *K-Nearest Neighbor*

K-Nearest Neighbor algorithm's main application is in the classification process. Integer k is chosen by KNN algorithms to divide the data from its closest neighbors.

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(pi - qi)^2} \quad (8)$$

A certain norm is used to determine the distance between the points. The new observation is assigned to the class that has the majority of the K nearest points. The norm is often used to determine how far apart two observations, q and p, are from one another. However, the observation is defined as Rn.

### *D. MLP*

A synthetic system having at least three layers of nodes is called a multilayer perceptron (hidden, input, and output). An encoder is used by each node. This enables scientists to choose which transistors should be ignored and deleted when building external networks.

$$\mathcal{X}_F^2 = \frac{12N}{K(K+1)}\left[\sum_j R_j^2 - \frac{K(K+1)^2}{4}\right] \quad (9)$$

where K is the total number of algorithms, N is the total number of datasets, and Rj is the average rank of algorithm j.

### *E. Logistic regression*

In a regression model called logistic regression, a categorical dependent variable is utilized, as the name suggests. The probability of a binary response using LR may be calculated using one or more independent variables. Predictions are transformed into probabilities using the sigmoid function.

$$y = w1 * x1 + w2 * x2 + \cdots + wn * xn + b \quad (10)$$

W1 to Wn are the weight vectors, and X1 displays the predicted result of logistic regression in the form of Y. The feature vector is Xn, and the bias is b.

### *A. Classification Models Comparison using novel K-CGAN model*

Table 3 shows the comparative results of novel K-CGAN with SMOTE, ADASYN, and B-SMOTE based on synthetic data. Table 3 demonstrates that based on the original data, all the classifier's performance is less than 82%.

TABLE III. COMPARISON OF CLASSIFICATION MODEL F1 SCORE USING K-CGAN

| Model | Original | SMOTE | ADASYN | B-SMOTE | K-CGAN |
|---|---|---|---|---|---|
| **XGBoost** | 81.321% | 99.941% | 99.853% | 99.812% | 99.951% |
| **Random forest** | 81.398% | 99.874% | 99.921% | 99.832% | 99.932% |
| **KNN** | 71.532% | 99.193% | 98.432% | 99.851% | 99.976% |
| **MLP** | 75.137% | 99.732% | 99.312% | 99.943% | 99.793% |
| **Logistic Regression** | 68.496% | 95.667% | 98.432% | 99.324% | 99.771% |

However, based on synthetic data SMOTE has 99.941% XGBoost, 99.874% Random forest, 99.193% KNN, 99.732% MLP, and 95.667% Logistic Regression. Further, ADASYN has 99.853% XGBoost, 99.921% Random Forest, 98.432% KNN, 99.312% MLP, and 98.432% Logistic Regression. Furthermore, B-SMOTE has 99.812% XGBoost, 99.832% Random Forest, 99.851% KNN, 99.943% MLP, and 99.324% Logistic Regression. Finally Novelty K-CGAN has 99.951% XGBoost, 99.932% Random forest, 99.976% KNN, 99.793% MLP, and 99.771% Logistic Regression.

## VI. RESULTS AND DISCUSSIONS

The collected data is divided into two parts, the first part is testing and the second one is training. The both data sets are dived with a specific ratio where 80% of class samples are for training set and 20% for testing set. The comparison between the classification techniques is calculated in the previous part of the study and data was collected for each classification method. The results between all these classifications are measured on the basis of the sensitivity, accuracy, precious, F-measure, specificity and also time taken.

As the tables in the experiment part of this study are showing that all the classification methods worked very well, with more than 0.90% in f1 score measure. The K-CGAN method produces higher results than any other resampling methods, as seen in Table 3. Therefore, these identified methods can be helpful when balancing an imbalanced dataset and identifying fraudulent and non-fraudulent transactions. Based on the evidence presented, K-CGAN is an effective resampling method to resolve credit card fraud issues.

## VII. CONCLUSIONS

Both XGBoost and Random Forest, two popular classification approaches, get almost similar results across all measures of performance, especially in the case of Novel test data augmentation method K-CGAN. Comparing the various class imbalance techniques, it is clear that oversampling accompanied by undersampling approaches may dramatically enhance the performance of the classifier due to their behavior. The research evaluates the performance of 5 classifiers in a credit card fraud detection using 4 class imbalance approaches. XGBoost and Random Forest, two popular ensemble classifiers, outperform KNN and MLP, two standard base classifiers, mostly because of their ability to work together. The Logistic Regression classifier had the worst performance across all class imbalance techniques. Future research milestone would be to evaluate K-CGAN with different datasets and compare its performance with XGBoost and Random Forest. This will provide more insight into the efficacy of class imbalance techniques such as oversampling when applied in combination with K-CGAN for novel test data augmentation. We anticipate this kind of analysis could help further refine and optimize the effectiveness of these machine learning approach for different datasets. Ultimately, this will improve our capability to accurately predict outcomes from data with minimal bias or errors.

## REFERENCES


[1] Statista, "Global fraud detection and prevention (FDP) market 2016-2022".,StatistaInc.https://www.statista.com/statistics/786778/worldwide-fraud-detection-and-prevention-market-size/ (accessed Dec. 22, 2022).

[2] Z. Chen, J. Duan, L. Kang, and G. Qiu, "A hybrid data-level ensemble to enable learning from highly imbalanced dataset," Inf. Sci. (Ny)., vol. 554, pp. 157–176, 2021.

[3] S. Negi, S. K. Das, and R. Bodh, "Credit Card Fraud Detection using Deep and Machine Learning," in 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2022, pp. 455–461.

[4] U. N. U. Mauludina, D. S. Y. Kartika, and A. D. M. Utomo, "A Fraud Detection Implementation Of Decision Tree C4. 5 Algorithm For Fraud Detection On Anonymous Credit Card Transaction," Int. J. Data Sci. Eng. Anaylitics, vol. 2, no. 2, pp. 16–23, 2022.

[5] P. Verma and P. Tyagi, "Credit Card Fraud Transaction Classification Using Improved Class Balancing and Support Vector Machines," in Recent Innovations in Computing, Springer, 2022, pp. 477–488.

[6] T. Chakraborty and A. K. Chakraborty, "Hellinger net: A hybrid imbalance learning model to improve software defect prediction," IEEE Trans. Reliab., vol. 70, no. 2, pp. 481–494, 2020.

[7] H. Ding, L. Chen, L. Dong, Z. Fu, and X. Cui, "Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection," Futur. Gener. Comput. Syst., vol. 131, pp. 240–254, 2022.

[8] Z. Li, M. Huang, G. Liu, and C. Jiang, "A hybrid method with dynamic weighted entropy for handling the problem of class imbalance with overlap in credit card fraud detection," *Expert Syst. Appl.*, vol. 175, p. 114750, 2021.

[9] H. Ding, Y. Sun, Z. Wang, N. Huang, Z. Shen, and X. Cui, "RGAN-EL: A GAN and ensemble learning-based hybrid approach for imbalanced data classification," *Inf. Process. Manag.*, vol. 60, no. 2, p. 103235, 2023.

[10] X. Wang, J. Xu, T. Zeng, and L. Jing, "Local distribution-based adaptive minority oversampling for imbalanced data classification," *Neurocomputing*, vol. 422, pp. 200–213, 2021.

[11] N. Huang *et al.*, "Multi-scale Interest Dynamic Hierarchical Transformer for sequential recommendation," *Neural Comput. Appl.*, pp. 1–12, 2022.

[12] J. Jedrzejowicz and P. Jedrzejowicz, "GEP-based classifier for mining imbalanced data," *Expert Syst. Appl.*, vol. 164, p. 114058, 2021.

[13] G. Wen, X. Li, Y. Zhu, L. Chen, Q. Luo, and M. Tan, "One-step spectral rotation clustering for imbalanced high-dimensional data," *Inf. Process. Manag.*, vol. 58, no. 1, p. 102388, 2021.

[14] H. Ba, "Improving detection of credit card fraudulent transactions using generative adversarial networks," *arXiv Prepr. arXiv1907.03355*, 2019.

[15] P. Vuttipittayamongkol and E. Elyan, "Neighbourhood-based undersampling approach for handling imbalanced and overlapped data," *Inf. Sci. (Ny).*, vol. 509, pp. 47–70, 2020.

[16] M. Manjurul Ahsan, M. Shahin Ali, and Z. Siddique, "Imbalanced Class Data Performance Evaluation and Improvement using Novel Generative Adversarial Network-based Approach: SSG and GBO," *arXiv e-prints*, p. arXiv-2210, 2022.

[17] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[18] S. Maldonado, C. Vairetti, A. Fernandez, and F. Herrera, "FW-SMOTE: A feature-weighted oversampling approach for imbalanced classification," *Pattern Recognit.*, vol. 124, p. 108511, 2022.

[19] J. Wei, H. Huang, L. Yao, Y. Hu, Q. Fan, and D. Huang, "NI-MWMOTE: An improving noise-immunity majority weighted minority oversampling technique for imbalanced classification problems," *Expert Syst. Appl.*, vol. 158, p. 113504, 2020.

[20] L. Abdi and S. Hashemi, "To combat multi-class imbalanced problems by means of over-sampling techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 238–251, 2015.

[21] B. Das, N. C. Krishnan, and D. J. Cook, "RACOG and wRACOG: Two probabilistic oversampling techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 1, pp. 222–234, 2014.

[22] X. Fan, X. Guo, Q. Chen, Y. Chen, T. Wang, and Y. Zhang, "Data augmentation of credit default swap transactions based on a sequence GAN," *Inf. Process. Manag.*, vol. 59, no. 3, p. 102889, 2022.

[23] S. Ger and D. Klabjan, "Autoencoders and generative adversarial networks for anomaly detection for sequences," *arXiv Prepr. arXiv1901.02514*, 2019.

[24] R. Zhang, Z. Zhang, and D. Wang, "RFCL: A new under-sampling method of reducing the degree of imbalance and overlap," *Pattern Anal. Appl.*, vol. 24, no. 2, pp. 641–654, 2021.

[25] X. Xie, H. Liu, S. Zeng, L. Lin, and W. Li, "A novel progressively undersampling method based on the density peaks sequence for imbalanced data," *Knowledge-Based Syst.*, vol. 213, p. 106689, 2021.

[26] Z. Nabulsi *et al.*, "Deep learning for distinguishing normal versus abnormal chest radiographs and generalization to two unseen diseases tuberculosis and COVID-19," *Sci. Rep.*, vol. 11, no. 1, pp. 1–15, 2021.

[27] E. Elyan, C. F. Moreno-Garcia, and C. Jayne, "CDSMOTE: class decomposition and synthetic minority class oversampling technique for imbalanced-data classification," *Neural Comput. Appl.*, vol. 33, no. 7, pp. 2839–2851, 2021.

[28] Z. Yu, K. Lan, Z. Liu, and G. Han, "Progressive ensemble kernel-based broad learning system for noisy data classification," *IEEE Trans. Cybern.*, 2021.

[29] K. Yang *et al.*, "Progressive hybrid classifier ensemble for imbalanced data," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 52, no. 4, pp. 2464–2478, 2021.

[30] W. W. Soh and R. M. Yusuf, “Predicting credit card fraud on a imbalanced data,” *Int. J. Data Sci. Adv. Anal. (ISSN 2563-4429)*, vol. 1, no. 1, pp. 12–17, 2019.

[31] S. Fu, X. Yu, and Y. Tian, “Cost sensitive ν-support vector machine with LINEX loss,” *Inf. Process. Manag.*, vol. 59, no. 2, p. 102809, 2022.

[32] S. Zhang, “Cost-sensitive KNN classification,” *Neurocomputing*, vol. 391, pp. 234–242, 2020.

[33] J. Kim, J. Kang, and M. Sohn, “Ensemble learning-based filter-centric hybrid feature selection framework for high-dimensional imbalanced data,” *Knowledge-Based Syst.*, vol. 220, p. 106901, 2021.

[34] J. Tian, P.-W. Tsai, F. Wang, K. Zhang, H. Xiao, and J. Chen, “An optional splitting extraction based gain-AUPRC balanced strategy in federated XGBoost for mitigating imbalanced credit card fraud detection,” *Int. J. Bio-Inspired Comput.*, vol. 20, no. 2, pp. 82–93, 2022.

[35] I. Sadgali, S. Nawal, and F. Benabbou, “Fraud detection in credit card transaction using machine learning techniques,” in *2019 1st International Conference on Smart Systems and Data Science (ICSSD)*, 2019, pp. 1–4.

[36] B. Gedela and P. R. Karthikeyan, “Credit Card Fraud Detection using AdaBoost Algorithm in Comparison with Various Machine Learning Algorithms to Measure Accuracy, Sensitivity, Specificity, Precision and F-score,” in *2022 International Conference on Business Analytics for Technology and Security (ICBATS)*, 2022, pp. 1–6.

[37] G. Ke *et al.*, “Lightgbm: A highly efficient gradient boosting decision tree,” *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[38] D. Chen, X.-J. Wang, C. Zhou, and B. Wang, “The distance-based balancing ensemble method for data with a high imbalance ratio,” *IEEE Access*, vol. 7, pp. 68940–68956, 2019.

[39] A. Singh, R. K. Ranjan, and A. Tiwari, “Credit card fraud detection under extreme imbalanced data: a comparative study of data-level algorithms,” *J. Exp. Theor. Artif. Intell.*, pp. 1–28, 2021.