Exploring Oversampling Techniques for Fraud Detection with Imbalanced Classes

***Abstract*—Credit card fraud has caused significant losses for financial institutions and individuals worldwide each year. Financial institutions must detect credit card fraud to prevent customers from being charged for products they did not order. Class imbalance has been a standing challenge for credit card transactions, as the number of fraudulent transactions is significantly lower than that of non-fraudulent transactions. In this paper, we comprehensive evaluate five oversampling techniques, namely Synthetic Minority Oversampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), Borderline SMOTE, Random Oversampling, and SMOTE Support Vector Machine (SMOTE SVM) in combination with seven machine learning techniques (namely XGBoost, Random Forest, K-Nearest Neighbor, Naive Bayes, Support Vector Machine, LightGBM, and Convolution Neural Network). Our results show oversampling generally improves fraud detection performance and SMOTE SVM is the better oversampling method than other methods under test. Notably, it achieved an accuracy of 76.47% when used with KNN on the smaller dataset and 99.93% with CNN on the larger dataset used in our experiments.**

***Index Terms*—Oversampling; Imbalanced Datasets; Classification; Fraud Detection; Machine Learning.**

I. Introduction

The rapid growth of financial institutions and the popularity of web-based e-commerce have contributed significantly to the growth of financial transactions in recent years. Fraud has become an increasingly severe issue due to the rise of online banking, especially the widespread use of cashless transactions. Credit card fraud may occur by fraudulently obtaining the credit card or gaining access to its information. When fraudsters use credit cards illegally, they make transactions without the permission of the cardholder [1]. Until now, credit card fraud has become a significant problem worldwide. In the single year of 2020, £783.8 million was reported as unauthorized, fraudulent transactions in the United Kingdom [2]. In 2021, over 703 fraudulent transactions have occurred in India [3].

While many strategies exist to resolve the problem and detect fraudulent transactions, fraudsters continually find new ways to obtain and exploit credit card information. Regardless of the cause of fraudulent transactions, a data-driven approach, represented by machine learning techniques, is desirable to analyze transaction histories automatically and detect fraudulent transactions effectively and efficiently. Detecting fraudulent transactions is equivalent to distinguishing legitimate from fraudulent transactions by learning patterns in transaction records and considering the transactions' contexts (e.g., during holidays and vacations) and changes in a customer's purchase behaviors [4].

Although many machine learning methods are available to the fraud detection problem, such as Random Forest (RF) [5], k-Nearest Neighbor (KNN) [6], Naive Bayes (NB) [7], Extreme Gradient Boosting (Xgboost) [8], Light Gradient Boosting Machine (LightGBM) [9], Convolutional Neural Networks (CNN) [10], and Support Vector Machine (SVM) [11], the imbalanced distribution of data poses challenges to the classification algorithms. In real-world applications, fraudulent transactions are rare compared to legitimate transactions, making the model biased towards the majority class, i.e., the training process spends most of its time on negative examples and does not learn enough from positive ones. It also risks viewing the minority label's transactions as outliers or noises during the training process [12], leading to degraded performance on the classification task.

While oversampling techniques are promising in rebalancing the numbers of samples in positive and negative classes [13], [14], there is still a lack of a comprehensive evaluation of the effectiveness of different oversampling techniques and machine learning methods to solve the fraud detection problem in various applications and business scenarios [15], [16]. In view of this knowledge gap, this paper applies a fraud detection process to two real-world datasets to evaluate the above techniques. In a nutshell, we make the following contributions:

- We explore the impact of imbalanced class distribution on two real-world datasets by comparing the performance of machine learning models before and after using oversampling techniques. The first dataset [17] has 167 samples with nine features, wherein 61 are fraudulent, and 106 are non-fraudulent transactions. The second dataset [18] is larger and more screwed, containing two days of fraudulent transactions by European cardholders. It is highly imbalanced, with the fraudulent transactions taking only 0.172% within.
- We test five oversampling techniques (namely SMOTE, SMOTE SVM, Random Oversampling, Borderline, and ADASYN) in combination with a series of classification techniques (namely RF, KNN, NB, SVM, Xgboost, LightGBM, and CNN) to validate the effectiveness of oversampling in fraud detection tasks. We also recommend promising oversampling techniques and machine-learning methods given specific tasks.

The paper is organized as follows. Section II introduces the related work. Section III overviews the oversampling methods and machine learning techniques under investigation for fraud

detection. Section IV reports our empirical studies to evaluate the above methods and discuss the results. Section V gives the concluding remarks.

II. Related Work

Many studies address the imbalanced distribution of data. Maniraj *et al.* [19] described different techniques to detect fraudulent transactions and the process to model credit card fraud detection. Due to the imbalanced nature of the dataset, the generated models achieved low precision values. Makki *et al.* [20] addressed the imbalanced issue through different experiments. Several solutions to this problem have been examined, and their weaknesses have been identified, mainly due to the number of false alarms. Using accuracy, sensitivity, and AUPRC as the measurement, they found that LR, C5.0 decision tree algorithm, SVM, and ANN were the most effective techniques.

Fraud detection in credit card transactions has been the subject of many research studies. Dal Pozzolo A *et al.* predicted results using ensemble models, incremental learning, and sampling techniques [21]. It was found that combining a synthetic minority oversampling approach (SMOTE) with a random forest classifier provided better results. Varmedja *et al.* [22] compared the performance of several algorithms, including RF, LR, Multilayer Perceptron and NB, to detect credit card fraud. SMOTE has also been used to solve the imbalanced dataset problem. The Random Forest algorithm provided the best results in accuracy and precision. Qaddoura *et al.* [23] Applied various oversampling techniques, such as SMOTE, ADASYN, borderline1 SMOTE and borderline2 SMOTE, and SMOTE-SVM. Also, different classification algorithms have been used to detect fraudulent transactions. The study found oversampling techniques improved the model's performance.

An imbalanced dataset may result in poor performance in machine learning applications, such as fraud detection. De *et al.* [24] investigated methods for optimizing supervised learning algorithms in such conditions, focusing on resampling. This study applied several ways to a spiral dataset with four classes: Gaussian Naive Bayes, Linear and Quadratic Discriminant Analysis, K-Nearest Neighbors, Support Vector Machine, Decision Trees, and Multi-Layer Perceptron. The oversampling technique achieves the best accuracy in the minority class, with a low number of false negatives at 99.928%. In the minority class, the results demonstrate that resampling strategies significantly improve model performance.

In [25], in sampling the data, they used SMOTE, Borderline-SMOTE and Adaptive Synthetic Sampling (ADASYN). Logistic Regression, Gradient Boosting, Random Forest, and XGboost have also been applied to the current public database on credit cards. As a result of the experiment, Gradient Boosting combined with ADASYN and SMOTE produced high accuracy totals of up to 99%. In [1], the light Gradient boosting machine algorithm is tuned by a Bayesian-based hyperparameter optimization technique, which utilizes an optimized lightGBM (light Gradient boosting machine). A 5-fold cross-validation test assesses the model's performance after selecting the most critical features using the Information Gain approach. As a result of the optimized light Gradient boosting algorithm, the accuracy, Area under Curve (AUC), and F1-Score were 98%, 0.9094, and 0.5695, respectively.

In [26], a convolutional neural network (CNN)-based approach is proposed in this study for detecting fraudulent transactions. Convolutional neural networks belong to deep learning and are feed-forward neural networks that combine more than one hidden layer. This paper proposes a new feature, trading entropy, to identify more complex fraud patterns and enhance classification accuracy. Using a cost-based sampling method, generating a significant number of frauds can alleviate the imbalance in a dataset with an imbalanced number of frauds. In this study, CNN has been used to detect fraud for the first time and has proven more accurate than other methods.

III. Materials and Methods

A. Dataset

In this study, two datasets were used:

- The first dataset used in this study was a small dataset named E-commerce Fraud, which is publicly available at [17]. The dataset consists of 167 samples, of which 61 are fraudulent, and 106 are non-fraudulent transactions.
- The second dataset [18] is known as the Credit Card Fraud Detection dataset. It consists of 284,808 records, 31 features, and a class identifying whether a record is a fraud. From 284,808 records, only 492 fraud records are found in this extremely imbalanced dataset.

The given datasets are imbalanced. Five oversampling techniques are employed: SMOTE, SMOTE SVM, Random Oversampling (ROS), Borderline, and ADASYN, which balance the dataset, making an equal number of fraudulent and non-fraudulent samples.

B. Preprocessing

An imbalanced dataset is oversampled in the preprocessing procedure, and different machine-learning techniques are applied. Following preprocessing, a ratio of 80:20 was used to divide the dataset into training and testing sets. In the next step, we applied oversampling to the training set. Several oversampling techniques exist, but only five are implemented in this paper: SMOTE, ADASYN, SMOTE SVM, Random Oversampling, and Borderline SMOTE.

After the oversampling step was implemented, seven models with or without oversampling were applied. It includes LightGBM, Xgboost, RF, KNN, NB, SVM, and CNN. To examine the effects of oversampling, these models were selected. Finally, different measures were employed to evaluate the system, including accuracy and F1-score, as shown in Figure 1.

C. Sampling Techniques

An imbalanced dataset can be dealt with using various techniques [27]. Four major methods are summarized and categorized: An algorithm-level, data-level, cost-sensitive learning, and an ensemble-based approach. Firstly, the algorithm-level

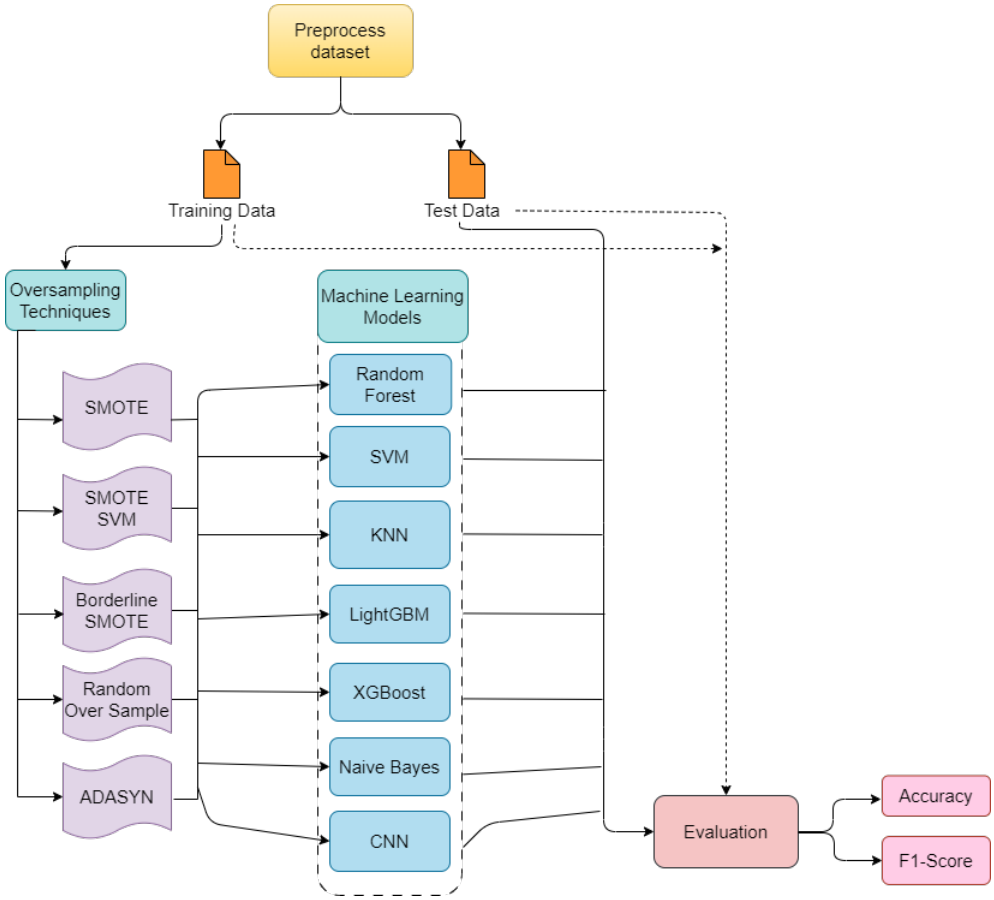

Fig. 1: Methodology

technique involves adapting existing learning algorithms for classifiers to bias the learning in favor of minorities. Second, the classification distribution is aimed to be rebalanced by the data level technique by resampling the data space. Next, a cost-sensitive learning technique is applied that optimizes the total cost errors for the two classes by combining data-level and algorithm-level techniques. Lastly, ensemble-based approaches combine a data-level and cost-sensitive algorithm with one of the earlier techniques.

Additionally, there are three types of data-level methods: undersampling, oversampling, and hybrid. The undersampling technique involves dropping the majority of instances of a class from the primitive dataset. This results in the loss of a great deal of highly valuable data from the source dataset. In contrast, using oversampling methods can result in duplicate instances in the minority class, increasing dataset size and increasing training time for machine learning algorithms. Hybrid sampling approaches combine both methods, which can be quite complicated. The oversampling techniques applied in this study include SMOTE, ADASYN, Borderline SMOTE, Random Oversampling, and SMOTE SVM algorithms to address these disadvantages. For solving skewed datasets, all of these techniques are commonly used and have demonstrated their effectiveness in a variety of applications [28], [29], [30], [31]. In the following 0sections, the oversampling techniques employed in this paper are explained in more detail.

1) Synthetic Minority Oversampling Technique (SMOTE) [32]: This widely used oversampling technique generates synthetic samples of the minority class by interpolating between existing minority class samples. Synthetic samples are created by selecting two existing minority class samples and generating a new sample that lies between them. This technique can produce high-quality synthetic samples but also increase the risk of overfitting if the synthetic samples are too similar to the existing minority class samples.

2) Adaptive Synthetic Sampling (ADASYN) [33]: This variant of SMOTE generates synthetic samples using a weighted distribution, where the weights are proportional to the density of the minority class samples. This can help produce more diverse synthetic samples, reducing the risk of overfitting.

3) Borderline SMOTE [34]: This is a variant of SMOTE that generates synthetic samples only for the minority class samples closest to the classifier's decision boundary. This can help improve the classifier's ability to classify samples near the border correctly.

4) Random Oversampling [32]: This method involves randomly selecting samples from the minority class and adding them to the original dataset. This technique can be effective but also introduce redundancy and overfitting if the same samples

are selected multiple times.

5) SMOTE SVM [35]: This technique generates new samples near the boundary of the decision. Like Borderline SMOTE, SMOTE SVM is based on the principle that generating new samples at the decision border is best. Using SMOTE SVM, decision boundaries are detected using support vectors.

D. Machine Learning Techniques

1) Support Vector Machine [11]: This technique involves locating observations at the boundary of each class. These observations are called support vectors. In supervised learning models, Support Vector Machines (SVM) are used to evaluate data used for classification and regression analysis. The SVM training technique allows one to create a model that categorizes new instances into two categories based on the training examples. Thus, the SVM is a nonprobabilistic binary linear classifier.

2) Random Forest [5]: It is one of several components of ensemble learning. Several decision trees constitute a forest; all their outputs are combined to form a class. Random Forest is used for classification and regression.

3) K-Nearest Neighbor [6]: In this technique, the classifier is presented with a new unknown sample. In this classifier, supervised learning is used to identify the K-nearest neighbors of the sample to identify areas of the pattern that belong to that neighbor's class. Then, the new pattern will be assigned accordingly, and the algorithm determines the proximity between two points based on their distance.

4) XGBoost [8]: The Extreme Gradient Boosting framework uses a gradient-boosted decision tree (GBDT) to enhance the performance of machine learning. It is one of the most widely used machine learning libraries for classification, regression, and ranking problems, in addition to parallel tree-boosting.

5) Light Gradient Boosting Machine (LightGBM) [9]: The technique is a type of gradient boosting based on decision trees. This method is used to enhance the efficiency of a classification model while consuming less memory. This technique performs various machine-learning applications, including ranking and classification. This method uses two techniques. The first method is known as Gradient-based One Side Sampling (GOSS), and the second method is known as Exclusive Feature Bundling (EFB), a technique developed to overcome the disadvantages of using the histogram approach in the Gradient Boosting Decision Tree (GDBT). EFB and GOSS methodologies are used to achieve the characteristics of the LightGBM model.

6) Naïve Bayes [7]: Naive Bayes is a representative supervised machine learning method for classification problems. The technique effectively estimates the parameters for classification using a small set of training data. By calculating the probability of the proper class, it uses Bayes' theorem to perform classification.

7) Convolution Neural Network [10]: The CNN contains several layers of interconnected neurons arranged into three types: convolutional, pooling, and fully connected layers. The first layer of a CNN involves a series of convolutions on the input image. These convolutions each involve a set of learnable filters that detect different features in the image. Convolutional layers generate output processed through nonlinear activation functions, such as ReLU, and then downsampled by pooling layers to reduce its dimensionality. The resulting feature maps are then processed through additional convolutional and pooling layers to extract increasingly complex features from the input image. As a final step, the feature maps have been flattened and passed through a layer or layers that are fully connected, which then performs the final classification or regression.

TABLE I: The confusion matrix

	Positive (Fraud)	**Negative (Legitimate)**
Positive (Fraud)	True Positive (TP)	False Negative (FN)
Negative (Legitimate)	False Positive (FP)	True Negative (TN)

IV. Result and Discussion

This section provides in-depth information about the performance metrics and results of the experiment. The last section discusses the results.

A. Evaluation Metrics

As illustrated in the table I, binary classification problems can be classified into four types based on how actual values are combined with predicted values: true positive, false positive, true negative, and false negative.

- True positive (TP): Represents the number of transactions that are expected to be fraudulent.
- False positive (FP): Represents the number of transactions believed to be fraudulent but legitimate.
- True negative (TN): Represents the number of samples predicted to be legal transactions and those legal transactions.
- False negative (FN): Represents the number of transactions believed to be legitimate but represent fraud [36].

The resampled datasets using the above oversampling techniques are each applied to seven machine-learning models implemented in Python. To assess the performance of the classifiers, accuracy and F1-score are used in this paper. Accuracy represents the percentage of predicting fraudulent and non-fraudulent classes correctly and is calculated as given in Eq. (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

F1-score is the harmonic mean of two other performance metrics, precision and recall, and is calculated as

$$F1\text{-}score = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{2}$$

TABLE II: Small Dataset Accuracies and F1-score of Various Classifiers before Oversampling Techniques

MODELS	ACCURACY (%)	F1-SCORE
Random Forest	**74.12**	**0.690**
SVM	47.94	0.250
KNN	60.76	0.425
XGBoost	65.76	0.607
Naïve Bayes	67.65	0.560
LightGBM	65.94	0.600
CNN	67.84	0.551

TABLE III: Large Dataset Accuracies and F1-score of Various Classifiers before Oversampling Techniques

MODELS	ACCURACY (%)	F1-SCORE
Random Forest	**99.74**	**0.913**
SVM	99.06	0.571
KNN	99.54	0.833
XGBoost	**99.77**	**0.923**
Naïve Bayes	98.30	0.510
LightGBM	**99.75**	**0.916**
CNN	99.70	0.904

B. Results

We analyzed two datasets by combining oversampling and machine learning techniques to identify how oversampling affects performance. Overall, our experimental results show that oversampling techniques improve the performance of machine learning models. However, different oversampling methods work differently with the various machine learning models. Therefore, it is important to choose the right oversampling technique with the correct machine-learning model to enhance the performance of the overall model. On both our experimental datasets, SMOTE SVM is the best oversampling method. It achieved the best performance when used with KNN and CNN on the two datasets, respectively.

We conducted experiments on a small dataset and a large dataset, respectively. Firstly, we tested seven machine learning techniques without oversampling for classification. The performance of each machine learning technique is shown in Table II and Table III. On the small dataset, Random Forest outperformed other classifiers; on the large dataset, XGBoost, Random Forest and LightGBM perform the best when compared with other classifiers.

In the second experiment, we oversampled each dataset using different methods and then trained machine learning models based on the oversampling results. Table IV and Table V show the performance metrics of each model under different oversampling methods on the small and large datasets, respectively. KNN with SMOTE SVM outperformed all the other machine learning models on the small dataset; while CNN also performed the best with SMOTE SVM–it was the best-performing model on the large dataset.

Besides, we conducted a third experiment to explore the models' stability based on standard deviations. We trained each model fifteen times on each dataset and presented their standard deviations in Table VI and Table VII. The results show that Random Forest and CNN exhibit non-zero standard deviations regardless of the oversampling techniques, meaning their performance could vary across multiple executions. In contrast, other methods (SVM, KNN, XGBoost, LightGBM and Naive Bayes) achieved zero or close-to-zero standard deviation, showing better stability in performance. We omit their standard deviations, which are extremely small and unsuitable to be presented in Table VI and Table VII.

V. Conclusion

Fraudulent credit card transactions are one of the most significant challenges that cause tremendous financial losses to businesses and individuals today. This paper evaluates the performance of various machine-learning algorithms with different oversampling techniques for fraudulent transaction detection. We empirically validate the effectiveness of incorporating oversampling techniques to overcome the class imbalance problem that exists extensively in real-world fraud detection datasets due to a lack of positive cases (i.e., fraudulent records). Besides, we achieve better performance than state-of-the-art solutions, reaching an accuracy of 76.47% and F1-score of 0.666 with KNN on one dataset, and an accuracy of 99.93% and F1-score of 0.857 with CNN on the other experimental dataset. In both settings, SMOTE SVM beats other oversampling methods and leads to superior results. Our next step is exploring swarm intelligence algorithms and stacked classifiers, which might further boost the performance.

References

[1] A. A. Taha and S. J. Malebary, "An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine," *IEEE Access*, vol. 8, pp. 25 579–25 587, 2020.

[2] F. T. F. 2021, "The definitive overview of payment industry fraud uk finance report," 2021. [Online]. Available: https://www.ukfinance.org.uk/policy-and-guidance/reports-publications/fraud-facts-2021

[3] T. Basuroy, "India: Number of credit and debit card frauds by leading state 2021," Oct 2022. [Online]. Available: https://www.statista.com/statistics/1097927/india-number-of-credit-debit-card-fraud-incidents-by-leading-state/

[4] "Combining unsupervised and supervised learning in credit card fraud detection," *Information Sciences*, vol. 557, pp. 317–331, 2021.

[5] A. Liaw, M. Wiener *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[6] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[7] G. I. Webb, E. Keogh, and R. Miikkulainen, "Naïve bayes." *Encyclopedia of machine learning*, vol. 15, pp. 713–714, 2010.

[8] 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.

[9] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *NIPS*, 2017.

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.

[12] R. Obiedat, R. Qaddoura, A. M. Al-Zoubi, L. Al-Qaisi, O. Harfoushi, M. Alrefai, and H. Faris, "Sentiment analysis of customers' reviews using a hybrid evolutionary svm-based approach in an imbalanced data distribution," *IEEE Access*, vol. 10, pp. 22 260–22 273, 2022.

TABLE IV: Small Dataset Accuracies and F1-scores of Various Classifiers for Different Oversampling Techniques. * denotes a significant improvement of a method over the method without an oversampling method applied (t-test $P < 0.05$).

Models	SMOTE		SMOTE SVM		Borderline		Random Oversampling		ADASYN	
	Accuracy (%)	F1-score	Accuracy (%)	F1-score	Accuracy (%)	F1-score	Accuracy (%)	F1-score	Accuracy (%)	F1-score
Random Forest	74.12	0.731*	71.76	0.677	72.55	0.694	73.14	0.686	65.14	0.618
SVM	58.82*	0.588*	65.12*	0.620*	62.50*	0.610*	65.70*	0.625*	62.70*	0.606*
XGBoost	61.76	0.580	70.59*	0.687*	64.71	0.600	58.82	0.533	64.71	0.625
KNN	67.65*	0.645*	**76.47***	**0.666***	70.59*	0.642*	67.65*	0.592*	61.76*	0.580*
LightGBM	61.76	0.580	64.71	0.571	68.50*	0.620*	61.76	0.551	61.76	0.551
Naïve Bayes	61.76	0.551	64.71	0.571	58.82	0.533	67.65	0.592	58.82	0.533
CNN	70.78*	0.669*	69.22*	0.598*	70.16*	0.632*	71.57*	0.646*	68.63*	0.616*

TABLE V: Large Dataset Accuracies and F1-scores of Various Classifiers for Different Oversampling Techniques. * denotes a significant improvement of a method over the method without an oversampling method applied (t-test $P < 0.05$).

Models	SMOTE		SMOTE SVM		Borderline		Random Oversampling		ADASYN	
	Accuracy (%)	F1-score	Accuracy (%)	F1-score	Accuracy (%)	F1-score	Accuracy (%)	F1-score	Accuracy (%)	F1-score
Random Forest	99.67	0.895	99.69	0.899	99.71	0.903	99.69	0.897	99.66	0.889
SVM	97.07	0.486	97.30	0.511	96.80	0.467	96.66	0.453	94.38	0.333
XGBoost	99.61	0.876	99.68	0.895	99.68	0.895	99.65	0.889	99.52	0.851
KNN	97.97	0.585	98.71	0.678	98.57	0.656	99.22	0.773	97.51	0.535
LightGBM	99.65	0.889	99.68	0.895	99.65	0.889	99.65	0.887	99.54	0.855
Naïve Bayes	97.14	0.500	96.87	0.477	96.57	0.454	97.12	0.498	95.81	0.405
CNN	99.80*	0.772	**99.93***	**0.857**	99.72	0.844	99.21	0.784	99.16	0.775

TABLE VI: The Standard Deviation of Accuracy after 15 Iterations on a Small Dataset.

Models	SMOTE	SMOTE SVM	Borderline	Random Oversampling	ADASYN
Random Forest	0.0369	0.0182	0.0200	0.0227	0.0227
CNN	0.0251	0.0237	0.0245	0.0232	0.0206

TABLE VII: The Standard Deviation of Accuracy after 15 Iterations on a Large Dataset.

Models	SMOTE	SMOTE SVM	Borderline	Random Oversampling	ADASYN
Random Forest	0.0002	0.0001	0.0002	0.0001	0.0002
CNN	0.0016	0.0011	0.0015	0.0016	0.0018

[13] R. Qaddoura, A. M. Al-Zoubi, I. Almomani, and H. Faris, "A multi-stage classification approach for iot intrusion detection based on clustering with oversampling," *Applied Sciences*, vol. 11, no. 7, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/7/3022

[14] R. Qaddoura, A. M. Al-Zoubi, H. Faris, and I. Almomani, "A multi-layer classification approach for intrusion detection in iot networks based on deep learning," *Sensors*, vol. 21, no. 9, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/9/2987

[15] J. Nanduri, Y.-W. Liu, K. Yang, and Y. Jia, "Ecommerce fraud detection through fraud islands and multi-layer machine learning model," in *Advances in Information and Communication: Proceedings of the 2020 Future of Information and Communication Conference (FICC), Volume 2*. Springer, 2020, pp. 556–570.

[16] I. Matloob, S. A. Khan, R. Rukaiya, M. A. K. Khattak, and A. Munir, "A sequence mining-based novel architecture for detecting fraudulent transactions in healthcare systems," *IEEE Access*, vol. 10, pp. 48 447–48 463, 2022.

[17] A. Rastogi, "Ecommerce fraud data," https://www.kaggle.com/datasets/aryanrastogi7767/ecommerce-fraud-data, May 2020.

[18] M. L. G. ULB, "Credit card fraud detection," https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud, Mar 2018.

[19] S. Maniraj, A. Saini, S. Ahmed, and S. Sarkar, "Credit card fraud detection using machine learning and data science," *International Journal of Engineering Research*, vol. 8, no. 9, pp. 110–115, 2019.

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

[21] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert systems with applications*, vol. 41, no. 10, pp. 4915–4928, 2014.

[22] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "Credit card fraud detection - machine learning methods," in *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, 2019, pp. 1–5.

[23] R. Qaddoura and M. M. Biltawi, "Improving fraud detection in an imbalanced class distribution using different oversampling techniques," in *2022 International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI)*, 2022, pp. 1–5.

[24] I. de Zarzà, J. de Curtò, and C. T. Calafate, "Optimizing neural networks for imbalanced data," *Electronics*, vol. 12, no. 12, p. 2674, 2023.

[25] H. Zou, "Analysis of best sampling strategy in credit card fraud detection using machine learning." New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3460179.3460186

[26] K. Fu, D. Cheng, Y. Tu, and L. Zhang, "Credit card fraud detection using convolutional neural networks," in *Neural Information Processing: 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16–21, 2016, Proceedings, Part III 23*. Springer, 2016, pp. 483–490.

[27] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from imbalanced data sets*. Springer, 2018, vol. 10.

[28] K. Li, W. Zhang, Q. Lu, and X. Fang, "An improved smote imbalanced data classification method based on support degree," in *2014 interna-*

tional conference on identification, information and knowledge in the internet of things. IEEE, 2014, pp. 34–38.

[29] L. Demidova and I. Klyueva, "Svm classification: Optimization with the smote algorithm for the class imbalance problem," in *2017 6th Mediterranean conference on embedded computing (MECO)*. IEEE, 2017, pp. 1–4.

[30] C. Lu, S. Lin, X. Liu, and H. Shi, "Telecom fraud identification based on adasyn and random forest," in *2020 5th International Conference on Computer and Communication Systems (ICCCS)*. IEEE, 2020, pp. 447–452.

[31] T. C. Tran and T. K. Dang, "Machine learning for prediction of imbalanced data: Credit fraud detection," in *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*. IEEE, 2021, pp. 1–7.

[32] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[33] 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)*, 2008, pp. 1322–1328.

[34] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I 1*. Springer, 2005, pp. 878–887.

[35] H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline over-sampling for imbalanced data classification," *International Journal of Knowledge Engineering and Soft Data Paradigms*, vol. 3, no. 1, pp. 4–21, 2011.

[36] D. Prusti and S. K. Rath, "Fraudulent transaction detection in credit card by applying ensemble machine learning techniques," in *2019 10th international conference on computing, communication and networking technologies (ICCCNT)*. IEEE, 2019, pp. 1–6.