mathematics

MDPI

Article

Estimating Financial Fraud through Transaction-Level Features and Machine Learning

Ayed Alwadain [1], Rao Faizan Ali [2] and Amgad Muneer [3,4,*]

[1] Computer Science Department, Community College, King Saud University, Riyadh 145111, Saudi Arabia
[2] Department of Software Engineering, School of Systems and Technology, University of Management and Technology, Lahore 54400, Pakistan
[3] Department of Imaging Physics, The University of Texas MD Anderson Cancer Center, Houston, TX 77030, USA
[4] Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar 32160, Malaysia
* Correspondence: muneeramgad@gmail.com or amabdulraheem@mdanderson.org

Abstract: In today's world, financial institutions (FIs) play a pivotal role in any country's economic growth and are vital for intermediation between the providers of investable funds, such as depositors, investors and users. FIs focus on developing effective policies for financial fraud risk mitigation however, timely prediction of financial fraud risk helps overcome it effectively and efficiently. Thus, herein, we propose a novel approach for predicting financial fraud using machine learning. We have used transaction-level features of 6,362,620 transactions from a synthetic dataset and have fed them to various machine-learning classifiers. The correlation of different features is also analysed. Furthermore, around 5000 more data samples were generated using a Conditional Generative Adversarial Network for Tabular Data (CTGAN). The evaluation of the proposed predictor showed higher accuracies which outperformed the previously existing machine-learning-based approaches. Among all 27 classifiers, XGBoost outperformed all other classifiers in terms of accuracy score with 0.999 accuracies, however, when evaluated through exhaustive repeated 10-fold cross-validation, the XGBoost still gave an average accuracy score of 0.998. The findings are particularly relevant to financial institutions and are important for regulators and policymakers who aim to develop new and effective policies for risk mitigation against financial fraud.

Keywords: financial fraud; transaction; machine learning; Conditional GAN; prediction; risk mitigation

MSC: 68T01

Citation: Alwadain, A.; Ali, R.F.; Muneer, A. Estimating Financial Fraud through Transaction-Level Features and Machine Learning. *Mathematics* **2023**, *11*, 1184. https://doi.org/10.3390/math11051184

Academic Editors: Javier Perote, Andrés Mora-Valencia and Trino-Manuel Ñíguez

Received: 22 January 2023
Revised: 20 February 2023
Accepted: 21 February 2023
Published: 28 February 2023

1. Introduction

Fraudulent transactions and their detectors have played complementary roles for a very long time. Fraudulent transactions are more common than ever, especially in the current Internet age, and they are the primary source of financial losses [1]. Around $28 billion was lost to the economy as a result of transaction fraud in 2019, $30 billion in 2020, and more than $32 billion in 2021. The number of fraudulent transactions worldwide is anticipated to increase yearly, reaching $34 billion in 2022, meaning that financial and banking service providers might need an automated fraud prevention tool [2].

Fraudsters in fraudulent transactions aim to deceive and steal from others to gain financial or other benefits. They often target vulnerable individuals or organizations and use tactics such as fake identities, phishing scams, or manipulating financial systems. The cost of payment fraud in developed countries, where electronic money dominates, is increasing, and the most common types of fraud are card-not-present and account takeover fraud [3,4]. To protect against fraud, individuals should be cautious with personal information, keep security systems up to date, and be vigilant with unsolicited requests. Financial institutions

are working to improve security and educate consumers, while governments and law enforcement are cracking down on fraudulent activities [5].

Systems for detecting fraud are designed to separate unusual activity patterns from a vast amount of transactional information, and then utilize those patterns to find or follow incoming transactions [6]. Artificial intelligence (AI) systems may automatically learn from their experiences and improve over time thanks to a technique called machine learning [7]. The creation of computer programs that can access information and utilize it to train themselves is the focus of machine learning [8]. The learning process starts with observations or data samples, including first-hand experience or instruction, so that we may search for patterns in the data and base future judgments on the examples we supply [9]. The main goal is to provide computers with the ability to learn from their experiences and modify their behaviour accordingly [10].

In artificial intelligence, deep learning (DL) is a subclass of machine learning (ML) that enables networks to learn independently using unstructured or unlabelled data. For example, by using profile photographs and identifying the variations between them, deep learning can be used to develop face detection and identify images as real or false [7]. Machine learning is quite effective in identifying and categorizing fraudulent transactions. In addition, a large volume of transaction records might be utilized to develop and test fraud classifiers [11]. Although supervised learning has been quite effective at identifying fraudulent transactions, transactional fraud analysis technology will continue to advance [12]. A corporation can save a substantial sum of money with a little classifier improvement [13].

There have been various studies proposed on fraudulent transactions. In a study in 2022 by Mosa et al. [14], a machine-learning-based method was proposed for predicting fraudulent financial transactions using 12 different algorithms, and it was observed that Random Forest (RF) outperformed all other classifiers with an accuracy of 99.97%; however, when a balanced dataset was used, the Bagging classifier showed an accuracy of 99.96%, outperforming all other methods.

According to a study reported in 2020 [15], people used cards more frequently than cash in their daily lives due to the rapid globalization of technology. MasterCard has evolved into a very practical piece of technology for online purchasing. This study used models of Recurrent Neural Networks (RNN), including long short-term memory (LSTM) and gated recurrent unit (GRU), as well as machine learning and deep learning techniques, to determine whether a transaction is legitimate or fraudulent. The suggested model outperformed the machine learning classifiers, scoring 91.37%, by a wide margin.

In the 2020 study by Lucas et al. [16], a unique ML algorithm was compared to identify fraud cases in addition to the automated classification and manual classification techniques being employed in fraud identification. For this, the authors employed the Support Vector Machine (SVM), Logistic Regression (LR), and RF. To advance a threat-scoring model, researchers learned about the targets. All algorithms underwent testing, and RF successfully outperformed other classifiers with the highest level of accuracy. Additionally, this approach was simple to learn and performed flawlessly on a sizable dataset. The outcome demonstrated how well this type of algorithm functions in the actual world.

The REDBSCAN method was employed in the study by Ge et al. in 2020 [17] to cut down on the number of samples while still maintaining the quality of the data. The area under the curve of support vector data description (SVDD) and SVM were compared and scored 94.60% and 97.75%, respectively. When SVDD was used in conjunction with REDBSCAN, it completed in 1.69 s, while it had taken 194 s to complete when it was used without REDBSCAN. The REDCSCAN algorithm produced better, quicker outcomes.

A hybrid approach was suggested by Yu et al. in 2020 [18] for detecting credit card fraud by combining the decision tree (DT) and Rough Set methods. Software programs, i.e., WEKA and MATLAB, were used throughout the entire study. The new strategy outperformed the old one with an 84.25% success rate after being used ten times. SVM was employed in the study by Dornadula and Geetha in 2019 [19] to categorize transactions as

legitimate or fraudulent. The SVM examined the cardholder's historical transaction patterns. It departed from its prior behaviour when a new transaction occurred by designating it as an illegal transaction. The best fraud detection score on SVM was 91%.

In a 2019 study by Thennakoon et al. [20], a deep learning-based approach to fraud detection was recommended. The log transformation was applied to deal with data skew issues in the dataset. The network makes use of focus reduction to train for challenging cases. The results demonstrated that the neural network model beat other, traditional models, such as SVM and LR. In 2018, Lakshmi and Kavila [21], measured the operation for detecting credit card fraud using the DT, LR, and RF algorithms. Oversampling was necessary since the dataset was quite imbalanced. A total of 60% of legal and 40% of illegal transactions are discovered after oversampling. These algorithms were implemented using the R language. DT's accuracy was 94.3%, RF's was 95.5%, and LR's was 90.0%. Measurements were also made for sensitivity, error rates, and specificity. Among these, the RF algorithm works admirably.

Some ML methods were suggested in the study by Carneiro et al. in 2017 [22] to assess the effectiveness of reasonably imbalanced data for fraud detection, as fraudulent samples were much less frequent than non-fraudulent ones. To assess the methods' potential, SVM, RF, DT, and LR were tested on raw and pre-processed data. These algorithms' respective accuracy levels were 97.5% (SVM), 98.6% (RF), 95.5% (DT), and 97.7% (LR). The RF excelled at handling enormous amounts of data, but it struggled with speed. A fraud detection system using LightGBM was proposed by Jain et al. in 2016 [23]. Comparisons with additional techniques, including LR, SVM, and XGBoost were conducted. In comparison to LR (92.60%), SVM (95.20%), and Xgboost (97.10%), LightGBM fared exceptionally well with an accuracy of 98%.

Seeja and Zareapoor in 2014 [24] primarily concentrated on a solution to the classification imbalance problem, and they looked into a machine learning algorithm-based fraud detection solution. They also discovered the shortcomings and condensed findings from their use of the labelled dataset for credit card fraud. They concluded that, when the data are significantly skewed, i.e., they deviant towards a direction instead of being uniformly distributed, the unbalanced classification is useless.

The present study aims to propose an accurate and reliable method for predicting fraudulent financial transactions using machine learning. To make the classifier more robust and accurate, we did not use any data balancing approach, as in real life, there are usually many more original transactions than fraudulent ones. Thus, we generated more samples of fraudulent transactions by using a conditional generative adversarial network for tabular data (CTGAN) and have further evaluated the potential of 27 machine learning classifiers. The method is proposed to help financial experts evaluate transactions accurately and efficiently rather than manually reviewing them through hectic procedures. The method is also important for financial institutions, regulators, and policymakers who aim to develop new and effective policies for risk mitigation against financial fraud.

2. Materials and Methods

In this study, we have performed an estimation of financial fraud through machine learning. By considering a synthetic dataset of financial transactions, we have further generated around 5000 samples using a conditional generative adversarial network for tabular data (CTGAN) [25]. Later, we trained 27 machine learning classifiers and performed an exhaustive performance evaluation of the best-performing model. A detailed elaboration on this method is provided in Figure 1.

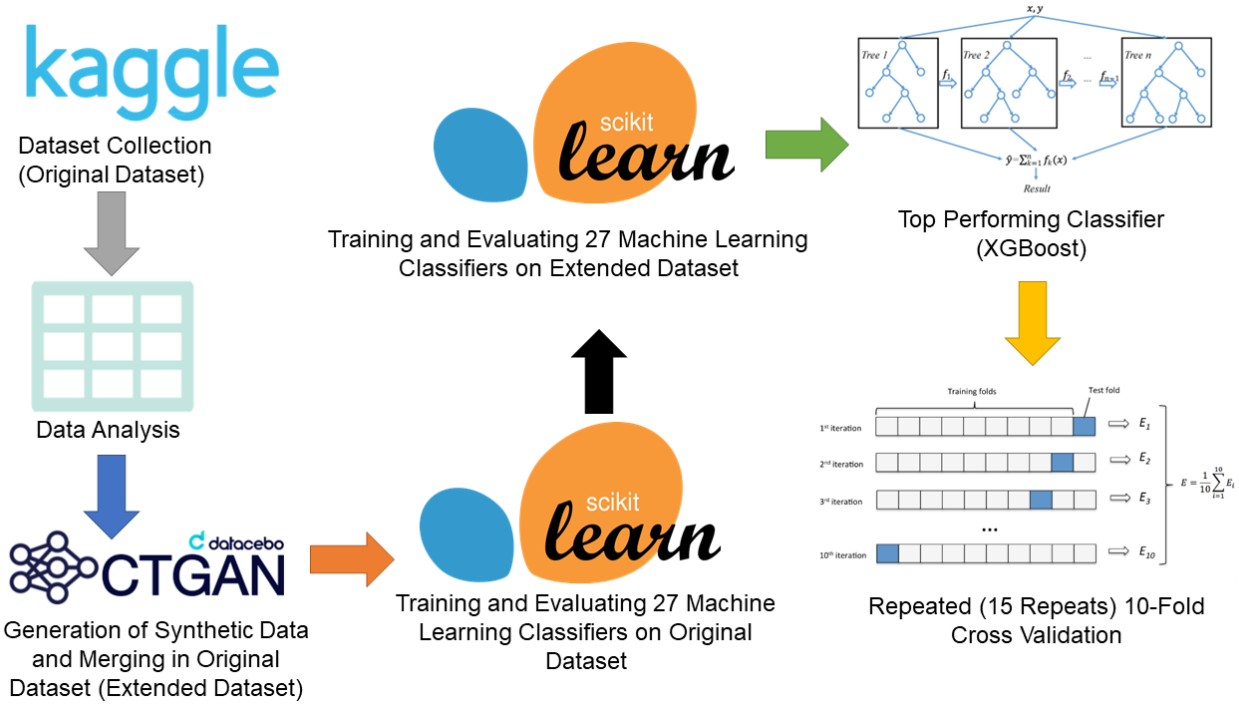


Figure 1. A graphical overview of the whole methodology.

2.1. Dataset Collection and Further Samples Generation using CTGAN

The dataset was taken from Kaggle Data Repository named "Synthetic Financial Datasets for Fraud Detection" (https://www.kaggle.com/datasets/ealaxi/paysim1 (accessed on 28 September 2022). This repository contains only 1/4th of the original dataset, as reported in [26,27], and comprises 6,362,620 samples, with 8213 fraudulent transactions and 6,354,407 non-fraudulent ones. The details of the variables are provided in Table S2. As the number of fraudulent transactions was much lower than the non-fraudulent ones, we generated 5000 samples of fraudulent transactions using CTGAN [25] to reduce the imbalance in the dataset; to do so, we used the Python implementation of CTGAN (https://github.com/sdv-dev/CTGAN (accessed on 5 October 2022). This resulted in a total of 13,213 fraudulent samples, and when added to the 6,354,407 non-fraudulent samples, the dataset now contained 6,367,620 samples in total. The remainder of the study was conducted using this dataset.

2.2. Data Analysis and Splitting Approaches

To understand the data further, we observed the correlation patterns among the features of the dataset. To do this, we plotted heatmap and feature-wise scatter/density plots for the whole dataset [28]. This helped us to understand the prominent features. After this analysis, the dataset was further split into training and testing groups with a ratio of 70:30, where 70% of the data (4,457,334 samples) was kept for training while 30% (1,910,286 samples) was used for testing. However, we also performed repeated (15 repeats) 10-fold cross-validation [29]; for that, we used the complete dataset of 6,367,620 samples, where data was split into 10 folds 15 times and was evaluated $10 \times 15 = 150$ times.

2.3. Machine Learning Classifiers

Using the data collected and pre-processed in the previous steps, we further passed it to 27 machine learning classifiers whose implementation was available in Scikit-Learn [30]. These included AdaBoost Classifier, Bagging Classifier, Bernoulli NB, Calibrated Classifier CV, Decision Tree Classifier, Dummy Classifier, ExtraTree Classifier, ExtraTrees Classifier, Gaussian NB, K Nearest Neighbors Classifier, Label Propagation, Label Spreading, LGBM Classifier, Linear Discriminant Analysis, Linear SVC, Logistic Regression, Nearest Cen-

troid, NuSVC, Passive Aggressive Classifier, Perceptron, Quadratic Discriminant Analysis, Random Forest Classifier, Ridge Classifier, Ridge Classifier CV, SGD Classifier, SVC, and XGBoost Classifier [31]. We performed the experiment in three different schemes. First, the original dataset of 6,362,620 samples was used to train and test the machine learning classifiers. All the classifiers were trained with their default implementations and parameters, and we evaluated the potential of machine learning classifiers for the original dataset. Later, we generated synthetic samples through CTGAN, and ran the experiment with an extended dataset of 6,367,620 samples, again with the default implementations and parameters. The reason for performing these two experiments was to evaluate the performance before and after the addition of synthetic samples.

Lastly, after these two experiments, we only considered the top-performing model, optimized the parameters using GridSearchCV, and evaluated it further on the whole dataset in an exhaustive manner [32]. We performed the repeated (15 repeats) 10-fold cross-validation for the complete dataset of 6,367,620 samples, where data was split into 10 folds 15 times and the top-performer classifier with optimized parameters was re-trained and evaluated $10 \times 15 = 150$ times.

2.4. Performance Evaluation

While proposing a novel predictor, it is really important to evaluate the performance, in terms of accuracy, effectiveness, efficiency and reliability. The evaluation method also has its own importance. In the present study, we performed 3 experiments where in the first experiment, we used the original dataset of 6,362,620 samples and performed the train–test split method to evaluate the performance. Here, the test split was considered an independent dataset because the samples of the test split were never used for any classifier during training.

For the second experiment, we used a similar approach to the train–test split, however, data from another 5000 fraudulent transactions were generated through CTGAN and added to the dataset for a total of 6,367,620 data samples. For the third experiment, we performed repeated 10-fold cross-validation for only the top-performing classifier, as this evaluation was already too exhaustive and performing this for multiple classifiers would have been very time consuming. For all these experiments, we evaluated performance using accuracy, the area under the receiver operating characteristic curve (AUC-ROC) and F1-Score metrics [33]. In addition, the time taken for training the 27 classifiers was computed for both experiments. The mathematical depiction for accuracy and F1-score is shown in Equations (1) and (2).

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

$$\text{F1} - \text{Score} = \frac{2\text{TP}}{(2\text{TP} + \text{FP} + \text{FN})}. \tag{2}$$

In Equations (1) and (2), the TP, TN, FP, and FN represent the true positive, true negative, false positive, and false negative, respectively. Here, true positive represents the number of fraudulent transactions which are correctly identified by the classifier as fraudulent. Similarly, true negative represents the number of those non-fraudulent transactions which are correctly identified by the classifier as non-fraudulent. As compared to this, false positive represents the number of those non-fraudulent transactions which are incorrectly identified by the classifier as fraudulent, while false negative represents the number of those fraudulent transactions which are incorrectly identified by the classifier as non-fraudulent [34]. The F1 score is an evaluation metric utilized to measure the performance of a classifier or model. It combines precision and recall in a harmonic mean, where precision evaluates the accuracy of positive predictions, and recall assesses the extent to which positive instances are captured by positive predictions. The F1 score is particularly useful in cases where the positive class is rare, or where there are significant consequences associated with false positive or false negative predictions. The score is expressed on a scale

of 0 to 1, with 1 signifying ideal precision and recall and 0 being the worst outcome [35]. The AUC-ROC was also considered a potential metric for performance evaluation and depicts the performance of classification problems at different threshold levels. The receiver operating characteristic curve is considered to be a probability curve, whereas the area under this curve (AUC-ROC) shows the classification capability of a classifier. The more area covered by the curve, the better the model's capability to distinguish classes; however, using the analysis from the present study, higher AUC-ROC shows the higher capability of classifiers for identifying fraudulent transactions.

3. Results

In this section, we discuss the results of our study. First, we analysed the data to understand patterns and correlations among different features in the dataset. Next, we implemented 27 machine learning classifiers, which are available in Scikit-Learn, and performed evaluations. We further extended our data by generating synthetic samples through CTGAN, then re-evaluated the 27 classifiers. Finally, for the top performing classifier, which was XGBoost in our study, we performed exhaustive evaluation through repeated 10-fold cross-validation. All experiments were conducted on a computer with an Intel Core i9-7920X CPU with 12 Cores, 64 GB of RAM, and an Nvidia RTX 2080Ti.

3.1. Analysis of Dataset

We analysed the dataset to study the patterns in features and also to identify the features with the most correlations. The purpose of performing this analysis was to observe if there are any prominent features and information in the dataset which can help in the classification of fraudulent and non-fraudulent transactions. Thus, firstly, the whole dataset was loaded and a correlation matrix in form of a heatmap was plotted using Matplotlib [36]. The heatmap is shown in Figure 2.

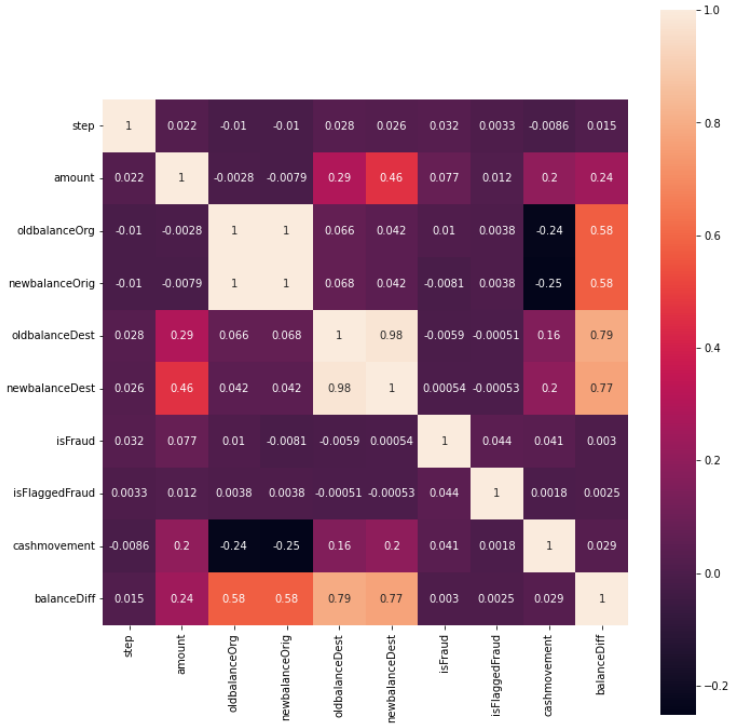


Figure 2. Correlation Matrix of Original Dataset.

Here, only those features were considered for plotting which have correlation values higher than 0.4. After plotting the correlation matrix as a heatmap, it was observed that, other than diagonal entries, i.e., correlation of a feature with itself, a few features had a very strong correlation, which were oldbalanceOrg, newbalanceOrg, oldbalanceDest and newbalanceDest. For further analysis, we plotted a scatter density plot to see the distribution of features (Figure 3).

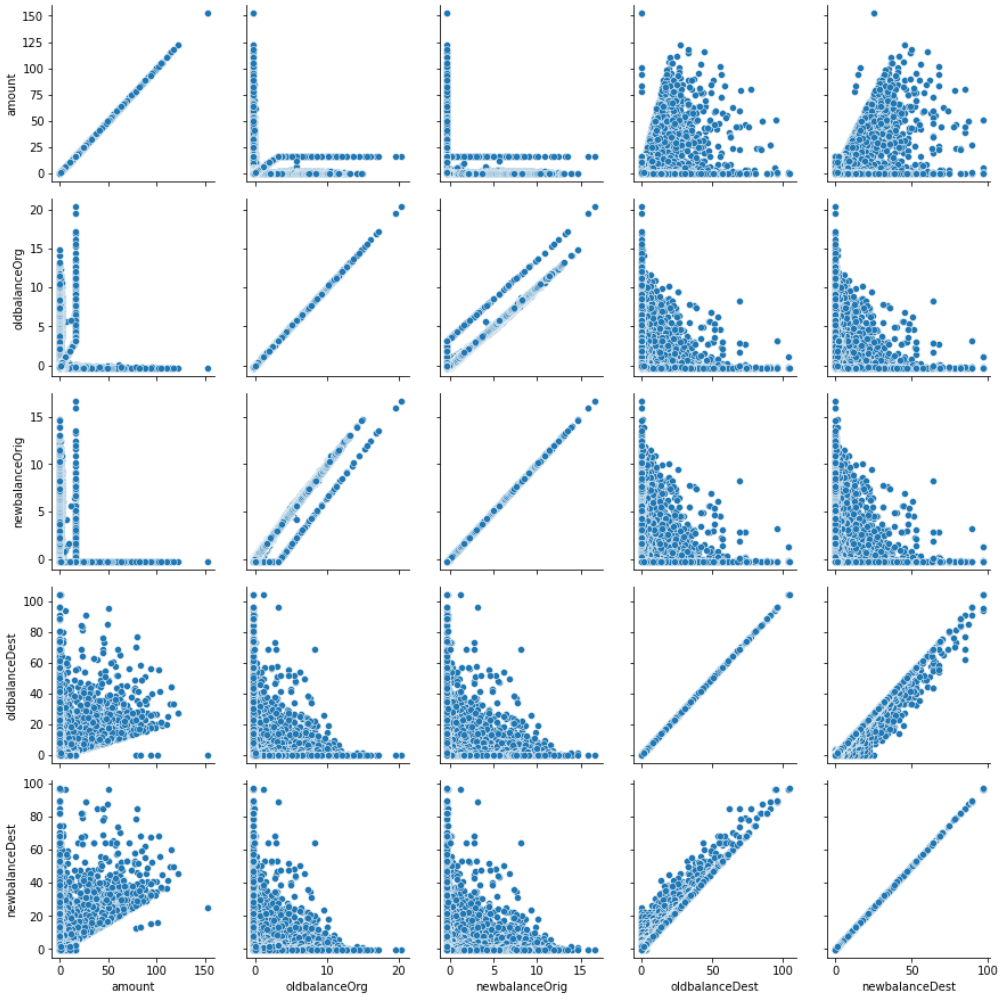


Figure 3. Scatter density plot for features of the dataset.

In Figure 3, the distribution of all features in the form of scatter plots can be observed; the figure also presents the correlation coefficients for all features.

3.2. Evaluation of Original Dataset

In this step, we used the original dataset to train 27 machine classifiers, which were discussed in Section 2.3. We used default implementations with default parameters for all classifiers and performed evaluations using an independent dataset that was created through the train–test split of the original dataset with a ratio of 70:30. The accuracy, AUC-ROC, and F1-score as computed for all classifiers are reported in Table 1.

Table 1. Evaluation of 27 Classifiers on the Original Dataset.

Model	Accuracy	AUC-ROC	F1-Score	Time Taken (s)
XGBClassifier	0.996	0.990	0.993	0.102
NuSVC	0.990	0.990	0.990	0.110
KNeighborsClassifier	0.960	0.960	0.960	0.025
ExtraTreesClassifier	0.930	0.930	0.930	0.271
LGBMClassifier	0.900	0.900	0.900	1.611
QuadraticDiscriminantAnalysis	0.900	0.900	0.900	0.034
SVC	0.890	0.890	0.890	0.639
RandomForestClassifier	0.890	0.890	0.890	0.706
LinearDiscriminantAnalysis	0.880	0.880	0.880	0.070
RidgeClassifierCV	0.880	0.880	0.880	0.065
RidgeClassifier	0.880	0.880	0.880	0.034
LinearSVC	0.880	0.880	0.880	0.298
CalibratedClassifierCV	0.880	0.880	0.880	1.012
LogisticRegression	0.870	0.870	0.870	0.035
AdaBoostClassifier	0.870	0.870	0.870	0.703
GaussianNB	0.840	0.840	0.840	0.019
SGDClassifier	0.820	0.820	0.820	0.040
BaggingClassifier	0.810	0.810	0.807	0.537
BernoulliNB	0.810	0.810	0.810	0.019
PassiveAggressiveClassifier	0.800	0.800	0.799	0.023
NearestCentroid	0.780	0.780	0.780	0.024
Perceptron	0.780	0.780	0.780	0.023
DecisionTreeClassifier	0.780	0.780	0.780	0.118
ExtraTreeClassifier	0.720	0.720	0.720	0.015
LabelSpreading	0.500	0.500	0.333	0.076
LabelPropagation	0.500	0.500	0.333	0.063
DummyClassifier	0.500	0.500	0.333	0.016

Through the evaluation provided in Table 1, it can be observed that the XGBoost classifier gave results with the highest accuracy and outperformed all other classifiers. The AUC-ROC for XGBoost was 0.990 while the F1-score was 0.993. The training of XGBoost was also performed very efficiently in 0.102 s.

3.3. Data Generation through CTGAN

To enhance the performance of our classifiers and to explore the potential of deep adversarial neural networks, we generated 5000 more synthetic samples of the fraudulent transaction using CTGAN. For this, the official implementation of CTGAN available at https://github.com/sdv-dev/CTGAN (accessed on 5 October 2022) was used. CTGAN was trained using an original dataset for 80 epochs only, as training this model was exhaustive and time consuming. However, as it was observed in 80 epochs, the model was well trained and the loss of training of the generator and discriminator model was minimized (Figure 4).

In Figure 4, it can be observed that training loss was minimized in 80 epochs only. Thus, it was shown that the CTGAN model learned well from the original dataset and would be able to produce the synthetic samples of fraudulent transactions accurately.

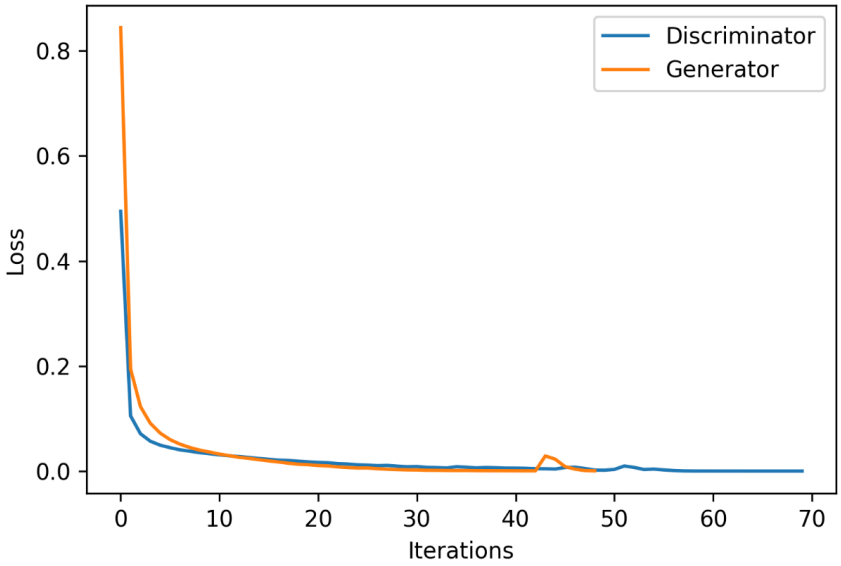


Figure 4. Loss of training of generator and discriminator model in CTGAN.

3.4. Evaluation of Updated Dataset

In this phase, we re-trained and evaluated the 27 machine learning classifiers using an extended dataset, to which we added 5000 more samples generated through CTGAN. Again, we used the same default implementations with default parameters for all classifiers and performed evaluation using an independent dataset, created again through the train–test split of the extended dataset with the ratio of 70:30. The accuracy, AUC-ROC, and F1-score were computed for all classifiers and are reported in Table 2.

Table 2. Evaluation of 27 Classifiers on Extended Dataset.

Model	Accuracy	AUC-ROC	F1-Score	Time Taken (s)
XGBClassifier	0.999	1.000	0.999	0.696
SVC	0.994	0.994	0.980	0.058
KNeighborsClassifier	0.993	0.993	0.980	0.021
RandomForestClassifier	0.990	0.984	0.980	0.711
NuSVC	0.980	0.974	0.980	0.106
LGBMClassifier	0.960	0.968	0.960	0.664
ExtraTreesClassifier	0.960	0.968	0.960	0.247
LinearDiscriminantAnalysis	0.960	0.947	0.960	0.051
RidgeClassifierCV	0.960	0.947	0.960	0.075
RidgeClassifier	0.960	0.947	0.960	0.032
BaggingClassifier	0.940	0.941	0.940	0.561
LogisticRegression	0.940	0.931	0.940	0.029
LinearSVC	0.940	0.931	0.940	0.287
CalibratedClassifierCV	0.940	0.931	0.940	1.072
NearestCentroid	0.920	0.915	0.920	0.023
GaussianNB	0.920	0.915	0.920	0.020
BernoulliNB	0.920	0.915	0.920	0.019
AdaBoostClassifier	0.920	0.915	0.920	0.722
SGDClassifier	0.900	0.899	0.900	0.095
PassiveAggressiveClassifier	0.880	0.883	0.881	0.020
Perceptron	0.880	0.883	0.881	0.015
QuadraticDiscriminantAnalysis	0.880	0.883	0.881	0.032
ExtraTreeClassifier	0.780	0.772	0.781	0.017
DecisionTreeClassifier	0.780	0.772	0.781	0.122
LabelSpreading	0.500	0.597	0.430	0.106
LabelPropagation	0.500	0.597	0.430	0.084
DummyClassifier	0.380	0.500	0.209	0.019

In this experiment, the XGBoost classifier again showed the highest accuracy and scores among all other metrics, however, the other classifiers shifted positions as compared to results reported for the previous experiment in Table 1. By analysing the results of Tables 1 and 2, it was concluded that XGBoost has a higher potential to identify fraudulent transactions than the other machine learning classifiers. Here, the accuracy of XGBoost increased further to 0.999 after adding the samples generated through CTGAN. After the addition, the AUC-ROC for XGBoost was 1.000 and the F1-score was 0.999. The training time of XGBoost did increase to 0.696 s, which is still quite efficient.

3.5. Repeated 10-Fold Cross-Validation

Considering the XGBoost classifier as the top-performing classifier, we further optimized its parameters using the GridSearchCV module of Scikit-Learn [37]. After fine-tuning, we applied an exhaustive evaluation method to thoroughly examine the potential of XGBoost for identifying fraudulent transactions. For this, we performed the repeated 10-fold cross-validation with 15 repeats for the extended dataset, where data was split into 10 folds, 15 times and the top-performer classifier, i.e., XGBoost with optimized parameters, was re-trained and evaluated $10 \times 15 = 150$ times (Table S1) [38]. Mean scores of 10 folds for each repeat is reported in Table 3 while a box plot is shown in Figure 5.

Table 3. Evaluation of XGBoost on Extended Dataset using repeated 10-fold Cross-Validation (Mean of 10 Folds for 15 repeats).

Repeat	Mean Accuracy	Mean AUC-ROC	Mean F1-Score
Repeat-1	0.999	0.998	0.998
Repeat-2	0.998	0.999	0.998
Repeat-3	0.999	0.999	0.998
Repeat-4	0.998	0.999	0.998
Repeat-5	0.998	0.998	0.999
Repeat-6	0.999	0.998	0.998
Repeat-7	0.999	0.998	0.998
Repeat-8	0.999	0.998	0.998
Repeat-9	0.999	0.998	0.999
Repeat-10	0.999	0.999	0.999
Repeat-11	0.998	0.999	0.998
Repeat-12	0.998	0.998	0.998
Repeat-13	0.997	0.998	0.999
Repeat-14	0.998	0.998	0.998
Repeat-15	0.998	0.998	0.998

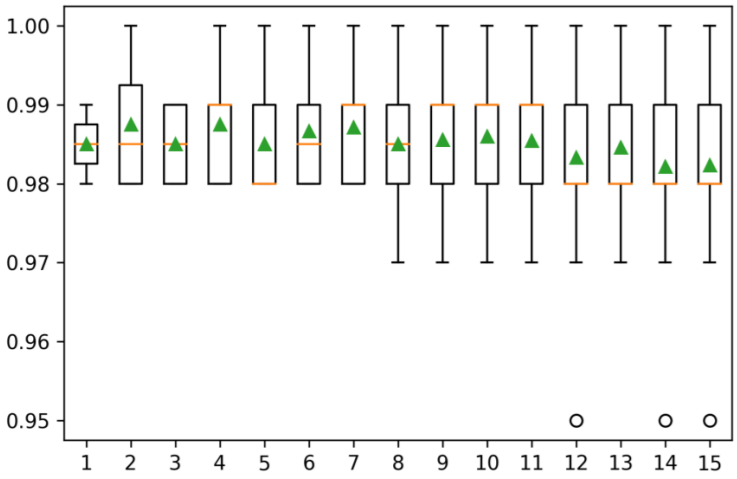


Figure 5. Boxplot for repeated 10-fold cross-validation for 15 repeats.

For all 15 repeats of 10-fold cross-validation, the performance of XGBoost remained persistent. In Figure 5, it is also evident that the performance of XGBoost was in the same range for all 10 folds of 15 repeats.

3.6. Final Evaluation of Original Dataset

Considering the XGBoost classifier as the top-performing classifier, and considering the final trained model, we performed predictions on the original dataset again to see the prediction capability of our model. The dataset comprised 6,362,620 samples, with 8213 fraudulent transactions and 6,354,407 non-fraudulent ones. By passing this original dataset into our model, we plotted the confusion matrix; the results are shown in Figure 6.

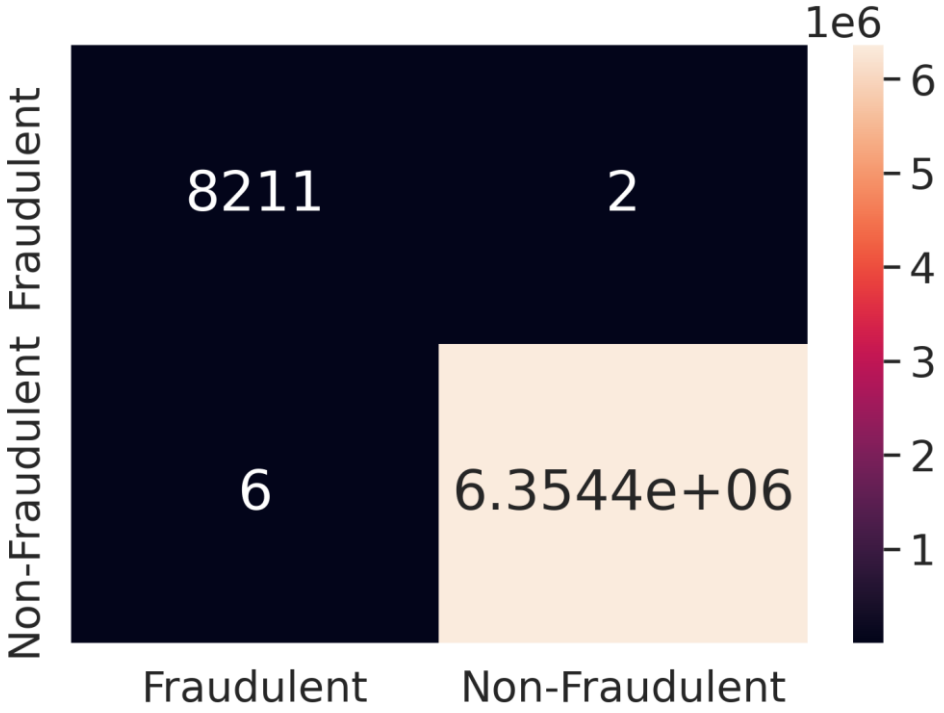


Figure 6. Confusion Matrix for Original Dataset on Final Model.

Furthermore, we computed the accuracy measures for this data. The results are shown in Table 4.

Table 4. Evaluation of Original Dataset on Final Model.

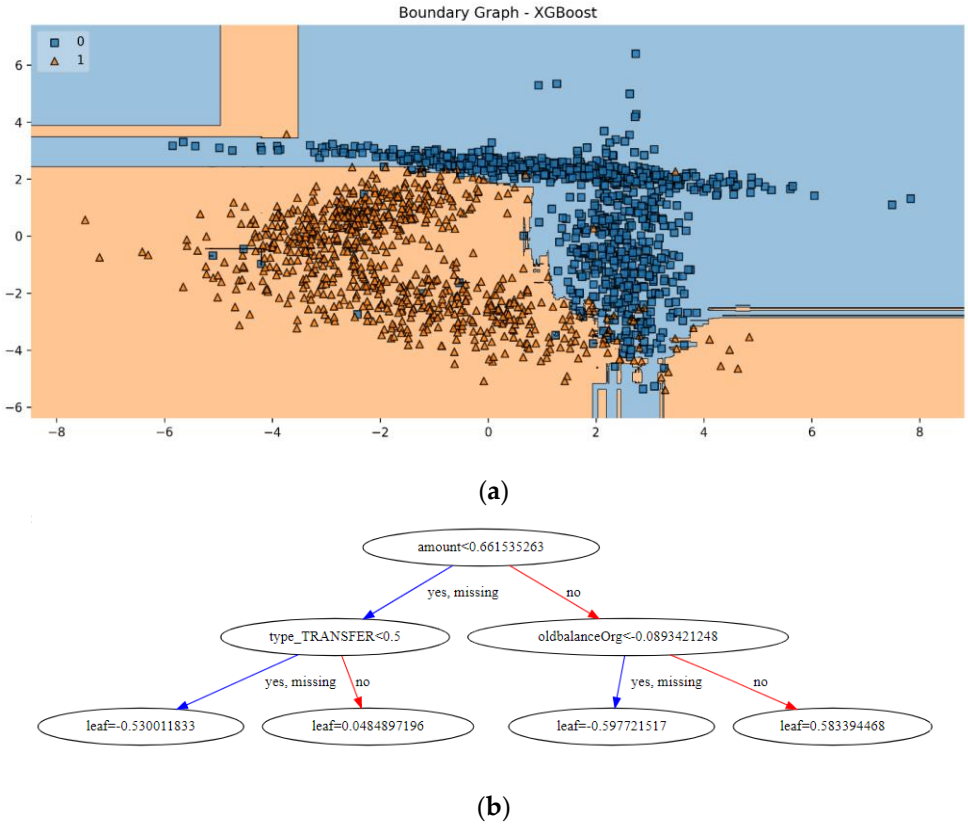
Model	Accuracy	AUC-ROC	F1-Score
XGBClassifier	0.999	1.000	0.999

The evaluation shows that, even on the original dataset, the final model has very high accuracy scores and correct prediction capability.

4. Discussion

In this study, we have proposed a novel machine-learning-based approach to identify fraudulent transactions using the dataset of transaction-level features. By considering a synthetic dataset of financial transactions, we have further generated around 5000 samples using a conditional generative adversarial network for tabular data (CTGAN). In our experiments, we first analysed the data to understand patterns and correlations among different features in the dataset. Next, we used the implementation of 27 machine learning classifiers, which are available in Scikit-Learn, and performed the evaluation. We further extended our data by generating synthetic samples through CTGAN and re-evaluated the

27 classifiers. Finally, for the top performing classifier, which was XGBoost in the present study, we performed exhaustive evaluation through repeated 10-fold cross-validation. To explore the classifying potential of XGBoost, we randomly selected a chunk of 100,000 records from our dataset and plotted the decision boundary of both XGBoost and Graphical Visualization, as shown in Figure 7.



(**a**)

(**b**)

Figure 7. Visualization of XGBoost (**a**) Decision Boundaries (**b**) Tree Visualization.

In Figure 7, the orange samples are the fraudulent transactions while the blue represents the non-fraudulent ones. The clear decision boundary can be seen in Figure 7, and the results of the present study are very good. A recent study by Mosa et al. [14], used the same original dataset as the present study and proposed a machine-learning-based method for predicting fraudulent financial transactions using 12 different algorithms. It was observed that, when a balanced dataset was used, the Bagging classifier showed an accuracy of 0.9996, outperforming all other methods. However, the authors used a synthetic method of data balancing, such as SMOTE (Synthetic Minority Over-sampling Technique), which is not a realistic approach in the real world [39]. In the real world, there are chances that there are a lower number of fraudulent transactions compared to non-fraudulent ones. In our study, we have not used such data balancing approaches and have achieved a consistent accuracy score of 0.999.

The proposed study is particularly relevant to financial institutions and is important for regulators and policymakers who aim to develop new and effective policies for risk mitigation against financial fraud.

5. Conclusions

With the recent advancement of technology, it has been observed that various tasks which are human dependent are now being performed by computational algorithms in an efficient and accurate manner. Machine learning and deep learning are known to have great potential to solve various problems in a very effective way. As financial institutions (FIs) play a pivotal role in any country's economic growth, these institutions emphasize their effective managerial policies to increase growth and overcome any issues being faced, such as risks. Mitigating risks is one of the main primary concerns of FIs, and one of the major risks to any such intuition is the risk of financial fraud. Keeping this in mind, we have proposed a novel approach for predicting financial fraud using machine learning. The proposed model contributes to empirical research about detecting fraud in financial transactions. A dataset with transaction-level features was analysed in this study, and 5000 more data samples were generated using CTGAN. Among the 27 classifiers being evaluated, XGBoost outperformed all other classifiers in terms of its accuracy score with 0.999 accuracies. When evaluated through exhaustive repeated 10-fold cross-validation, the XGBoost still gave an average accuracy score of 0.998. We have proposed this model by considering the importance of financial institutions, as it will help financial experts to evaluate transactions accurately and efficiently, instead of using manual, hectic procedures. This information is also important for regulators and policymakers who aim to develop new and effective policies for risk mitigation against financial fraud. By increasing the data regarding fraudulent transactions, the model can be made even more reliable; however, using the current dataset, the proposed model has shown very accurate, efficient, and effective results.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/math11051184/s1, Table S1: Detailed results of repeated 10-fold cross-validation for 15 repeats. Table S2: Details of variables.

Author Contributions: Conceptualization, A.A. and A.M.; methodology, A.A.; software, A.M.; validation, A.A., R.F.A. and A.M.; formal analysis, A.A.; investigation, A.M.; resources, A.M.; data curation, A.A. and A.M; writing—original draft preparation, A.A. and R.F.A.; writing—review and editing, A.M.; visualization, A.A. and R.F.A.; supervision, A.A. and A.M.; project administration, A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by Researchers Supporting Project number (RSP2023R309), King Saud University, Riyadh, Saudi Arabia.

Data Availability Statement: The dataset is available at Kaggle in a Repository named "Synthetic Financial Datasets for Fraud Detection" (https://www.kaggle.com/datasets/ealaxi/paysim1 (accessed on 28 September 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kyriienko, O.; Magnusson, E.B. Unsupervised quantum machine learning for fraud detection. *arXiv* **2022**, arXiv:2208.01203.
2. Kulatilleke, G.K. Challenges and complexities in machine learning based credit card fraud detection. *arXiv* **2022**, arXiv:2208.10943.
3. Levi, M.; Burrows, J.; Fleming, M.; Hopkins, M.; Matthews, K.G.P. *The Nature, Extent and Economic Impact of Fraud in the UK*; Association of Chief Police Officers (ACPO): Mays Landing, NJ, USA, 2007.
4. Van Driel, H. Financial fraud, scandals, and regulation: A conceptual framework and literature review. *Bus. Hist.* **2018**, *61*, 1259–1299. [CrossRef]
5. Okoye, E.I.; Gbegi, D.O. An evaluation of the effect of fraud and related financial crimes on the Nigerian economy. *Kuwait Chapter Arab. J. Bus. Manag. Rev.* **2013**, *33*, 1–23. [CrossRef]
6. Aziz, R.M.; Baluch, M.F.; Patel, S.; Ganie, A.H. LGBM: A machine learning approach for Ethereum fraud detection. *Int. J. Inf. Technol.* **2022**, *14*, 3321–3331. [CrossRef]
7. Ahmed, S.; Alshater, M.M.; El Ammari, A.; Hammami, H. Artificial intelligence and machine learning in finance: A bibliometric review. *Res. Int. Bus. Financ.* **2022**, *61*, 101646. [CrossRef]
8. Alfaiz, N.S.; Fati, S.M. Enhanced Credit Card Fraud Detection Model Using Machine Learning. *Electronics* **2022**, *11*, 662. [CrossRef]

9. Aziz, S.; Dowling, M.; Hammami, H.; Piepenbrink, A. Machine learning in finance: A topic modeling approach. *Eur. Financ. Manag.* **2022**, *28*, 744–770. [CrossRef]
10. Chaquet-Ulldemolins, J.; Gimeno-Blanes, F.-J.; Moral-Rubio, S.; Muñoz-Romero, S.; Rojo-Álvarez, J.-L. On the Black-Box Challenge for Fraud Detection Using Machine Learning (I): Linear Models and Informative Feature Selection. *Appl. Sci.* **2022**, *12*, 3328. [CrossRef]
11. Bertucci, L.; Briere, M.; Fliche, O.; Mikael, J.; Szpruch, L. Deep Learning in Finance: From Implementation to Regulation. SSRN 4080171. 2022. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4080171 (accessed on 8 January 2022).
12. D'Amato, V.; Levantesi, S.; Piscopo, G. Deep learning in predicting cryptocurrency volatility. *Phys. A Stat. Mech. Appl.* **2022**, *596*, 127158. [CrossRef]
13. Saheed, Y.K.; Baba, U.A.; Raji, M.A. Big Data Analytics for Credit Card Fraud Detection Using Supervised Machine Learning Models. In *Big Data Analytics in the Insurance Market*; Emerald Publishing Limited: Bingley, UK, 2022; pp. 31–56.
14. Megdad, M.M.; Abu-Naser, S.S.; Abu-Nasser, B.S. Fraudulent Financial Transactions Detection Using Machine Learning. *Int. J. Acad. Inf. Syst. Res. (IJAISR)* **2022**, *6*, 30–39.
15. Khedmati, M.; Erfani, M.; GhasemiGol, M. Applying support vector data description for fraud detection. *arXiv* **2020**, arXiv:2006.00618.
16. Lucas, Y.; Portier, P.-E.; Laporte, L.; He-Guelton, L.; Caelen, O.; Granitzer, M.; Calabretto, S. Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs. *Future Gener. Comput. Syst.* **2020**, *102*, 393–402. [CrossRef]
17. Ge, D.; Gu, J.; Chang, S.; Cai, J. Credit card fraud detection using lightgbm model. In Proceedings of the 2020 international conference on E-commerce and internet technology (ECIT), Zhangjiajie, China, 24–26 April 2020; pp. 232–236.
18. Yu, X.; Li, X.; Dong, Y.; Zheng, R. A deep neural network algorithm for detecting credit card fraud. In Proceedings of the 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Fuzhou, China, 12–14 June 2020; pp. 181–183.
19. Dornadula, V.N.; Geetha, S. Credit card fraud detection using machine learning algorithms. *Procedia Comput. Sci.* **2019**, *165*, 631–641. [CrossRef]
20. Thennakoon, A.; Bhagyani, C.; Premadasa, S.; Mihiranga, S.; Kuruwitaarachchi, N. Real-time credit card fraud detection using machine learning. In Proceedings of the 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 10–11 January 2019; pp. 488–493.
21. Lakshmi, S.; Kavilla, S.D. Machine learning for credit card fraud detection system. *Int. J. Appl. Eng. Res.* **2018**, *13*, 16819–16824.
22. Carneiro, N.; Figueira, G.; Costa, M. A data mining based system for credit-card fraud detection in e-tail. *Decis. Support Syst.* **2017**, *95*, 91–101. [CrossRef]
23. Jain, R.; Gour, B.; Dubey, S. A hybrid approach for credit card fraud detection using rough set and decision tree technique. *Int. J. Comput. Appl.* **2016**, *139*, 1–6. [CrossRef]
24. Seeja, K.; Zareapoor, M. Fraudminer: A novel credit card fraud detection model based on frequent itemset mining. *Sci. World J.* **2014**, *2014*, 252797. [CrossRef]
25. Xu, L.; Skoularidou, M.; Cuesta-Infante, A.; Veeramachaneni, K. Modeling tabular data using conditional gan. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–11.
26. Lopez-Rojas, E.; Elmir, A.; Axelsson, S. PaySim: A financial mobile money simulator for fraud detection. In Proceedings of the 28th European Modeling and Simulation Symposium, EMSS, Larnaca, Cyprus, 26–28 September 2016; pp. 249–255.
27. Lopez-Rojas, E.A. *Applying Simulation to the Problem of Detecting Financial Fraud*; Blekinge Tekniska Högskola: Karlskrona, Sweden, 2016.
28. Archakov, I.; Hansen, P.R. A new parametrization of correlation matrices. *Econometrica* **2021**, *89*, 1699–1715. [CrossRef]
29. Kim, J.-H. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Comput. Stat. Data Anal.* **2009**, *53*, 3735–3745. [CrossRef]
30. Hao, J.; Ho, T.K. Machine learning made easy: A review of scikit-learn package in python programming language. *J. Educ. Behav. Stat.* **2019**, *44*, 348–361. [CrossRef]
31. Pölsterl, S. scikit-survival: A Library for Time-to-Event Analysis Built on Top of scikit-learn. *J. Mach. Learn. Res.* **2020**, *21*, 8747–8752.
32. Rtayli, N.; Enneya, N. Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization. *J. Inf. Secur. Appl.* **2020**, *55*, 102596. [CrossRef]
33. Hossin, M.; Sulaiman, M.N. A review on evaluation metrics for data classification evaluations. *Int. J. Data Min. Knowl. Manag. Process* **2015**, *5*, 1.
34. Marom, N.D.; Rokach, L.; Shmilovici, A. Using the confusion matrix for improving ensemble classifiers. In Proceedings of the 2010 IEEE 26-th Convention of Electrical and Electronics Engineers, Eilat, Israel, 17–20 November 2010; pp. 000555–000559.
35. Lipton, Z.C.; Elkan, C.; Narayanaswamy, B. Thresholding classifiers to maximize F1 score. *arXiv* **2014**, arXiv:1402.1892.
36. Barrett, P.; Hunter, J.; Miller, J.T.; Hsu, J.-C.; Greenfield, P. matplotlib—A Portable Python Plotting Package. In Proceedings of the Astronomical data analysis software and systems XIV, Pasadena, CA, USA, 24–27 October 2005; p. 91.

37. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
38. Chen, T.; He, T.; Benesty, M.; Khotilovich, V.; Tang, Y.; Cho, H.; Chen, K. Xgboost: Extreme gradient boosting. *R Package Version 0.4-2* **2015**, *1*, 1–4.
39. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]