IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 September 2023, accepted 23 September 2023, date of publication 27 September 2023,
date of current version 5 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3320072

RESEARCH ARTICLE

Generative Adversarial Networks-Based Novel Approach for Fraud Detection for the European Cardholders 2013 Dataset

FAHDAH A. ALMARSHAD[1], GHADA ABDALAZIZ GASHGARI[2], AND ABDULLAH I. A. ALZAHRANI[3]

[1]Department of Information Systems, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al Kharj 11942, Saudi Arabia
[2]Department of Cybersecurity, College of Computer Science and Engineering, University of Jeddah, Al Faisaliyyah, Jeddah 23445, Saudi Arabia
[3]Department of Computer Science, College of Science and Humanities in Al Quwaiiyah, Shaqra University, Shaqra 11961, Saudi Arabia

Corresponding author: Fahdah A. Almarshad (f.almarshad@psau.edu.sa)

This work was supported by the Deputyship for Research and Innovation, Ministry of Education, Saudi Arabia, under Project IF2/PSAU/2022/01/21905.

ABSTRACT Credit card use poses a significant security issue on a global scale, with rule-based algorithms and traditional anomaly detection being two of the most often used methods. However, they are resource-intensive, time-consuming, and erroneous. Given fewer instances than legal payments, the dataset imbalance has become a serious issue. On the other hand, the generative technique is considered an effective way to rebalance the imbalanced class issue, as this technique balances both minority and majority classes before the training. In a more recent period, GAN is considered one of the most popular data generative techniques, as it is used in significant data settings. Hence, the research under study explores a classification system to detect fraudulent credit card transactions that are being trained using the European Cardholders 2013 dataset. It has 30 features, 28 of which are hidden due to sensitive information. Fraud activity accounts for less than 1% of the entire transaction volume of $284807. Additionally, GANs is a generative model based on game theory, in which a generator G and a discriminator D compete with one another. The generator's goal is to make the discriminator uncertain. Distinguishing between instances from the generator and those from the original dataset is the discriminator's goal, and we can increase classifiers' discriminating strength by training GANs on a set of fraudulent credit card transactions. According to the outcome, our model outperformed the earlier experiments with an AUC score of 0.999. Additionally, it creates artificial data using GANs, enabling the production of a sizable volume of high-quality data. In terms of innovation and performance, this technique substantially improves over earlier research.

INDEX TERMS Credit card fraud detection, imbalanced data, generative adversarial networks, deep learning, fraud detection.

I. INTRODUCTION

Cyber fraud involving credit cards is a significant problem for the banking sector, resulting in the loss of billions of dollars yearly [1]. The banking sector needs to work on improving cyber security [2]. For detecting and tracking credit card fraud, several technologies have been created [3]. However, due to the continuously changing nature of threats, the banking sector has to be outfitted with the most cutting-edge and efficient cyber fraud management systems [3]. In recent years, the use of credit cards and other online payment methods has skyrocketed, which has increased the prevalence of credit card cybercrime. There are several types of credit card fraud. For instance, a credit card gets stolen, or cyber fraud involves the theft of private credit card information [4].

Furthermore, inputting data during an online purchase increases the risks of fraud. However, machine learning (ML) scholars have been interested in the difficult job of detecting

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

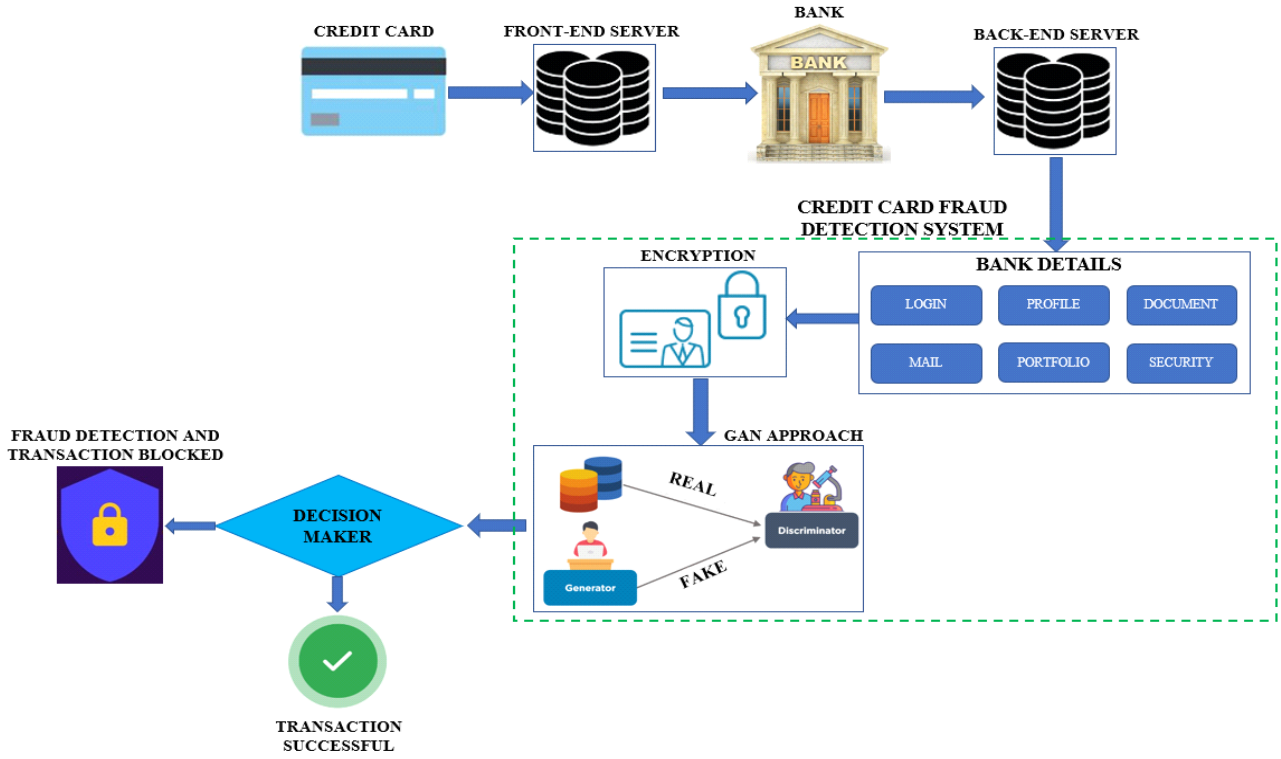

FIGURE 1. Framework of credit card fraud detection using GANs.

cyber theft on credit cards [3], [5]. The skewness of credit card-related datasets is rather pronounced. When dealing with datasets that have a significant skew, a large number of algorithms find it difficult to distinguish between objects from minority classes. Systems designed to detect cyber fraud must act quicker to be effective [6]. The impact of new attack techniques on the conditional distribution of the data across the period is another significant area of worry.

Fraud detection is typically viewed as a pattern classification issue that distinguishes aberrant patterns from normal patterns. For this purpose, machine learning (ML), data mining, and traditional statistical classification approaches have all been used effectively [7]. Credit card fraud has been successfully detected using a variety of artificial neural network (ANN) models [8], which, by replicating the characteristics of interacting neurons, are renowned for their ability to simulate extremely non-linear and complex functions from the ground up [9]. Additionally, studies have been conducted utilizing explicit entity-relation networks to identify probable fraud. To find anomalies, the majority of data mining and machine learning approaches significantly rely on enormous volumes of transactional or operational data. Data from the operational database of the telecommunication provider, including average call time, call volume, and caller location, is widely used by tools for telecom fraud detection. For instance, it is possible to identify credit card theft by contrasting questionable transactions with information extracted from long-term history data about previous client usage trends [10], [11]. In addition, classification models like the multi-layer perceptron (MLP) [12], random forest (RF) [13], and logistic regression (LR) [14] are frequently discriminative models [15], which means they aim to identify the most suitable class by utilizing a certain feature set. It is the most effective underlying cause of the bias issue brought on by the data imbalance, as the algorithm does not have a concept of "how" the data are created, focusing instead on the objective measure of discriminating [11]. One technique to solve this issue is using models that seek to comprehend the underlying creative process by generative networks. With this goal in mind, Gaussian Mixture Models (GMMs) are the foundation for many generative models, including Hidden Markov Models [16]. Still, their use in classification tasks is labor-intensive and requires the assumption of Gaussian distributions [17]. To provide the necessary classification procedure, such models have been combined with clustering methods [18].

Figure 1 depicts the framework for Card fraud detection using the Generative Adversarial Network.

In addition, oversampling the minority class is one of the primary causes of false data generation. To get knowledge from the imbalanced datasets, oversampling is employed.

The imbalance issue still exists in several real-world datasets. The small number of illegal actions compared to

legitimate ones in the credit card fraud field creates an unequal distribution of wealth [19]. The classification algorithms find it challenging to recognize fraudulent transactions in this situation [20]. A solution to the issue of the unequal distribution of class is to augment the minority class statistics. Instances with characteristics resembling those of the original data are created to achieve this goal. In addition to preventing overfitting, augmentation aids in averting the underrepresentation of the minority class [21]. Avoiding governmental restrictions and preserving client privacy are two other justifications for creating false datasets. Using financial data, such as credit card transaction data, as an example, might put consumers' privacy in danger because it contains sensitive information about them. Using synthetic datasets to train the model is one method for addressing privacy issues [22].

One of the most well-known generating approaches is generative adversarial networks (GANs), which were created by [23] and [24]. This work examines the adaptability and scalability of GANs to produce fake samples for credit card fraud detection. To expand the body of knowledge on GANs for data augmentation in unbalanced classes, this work attempts to add to the current body of knowledge. When it comes to solving unbalanced class problems, researchers [23] contend that employing GANs is the most appropriate and efficient strategy compared to other machine learning approaches. Due to its versatility and ability to comprehend hidden data structures, it is also extremely resilient against overlapping and overfitting. To do this, we looked at earlier investigations by scientists working in fraud detection who enhanced credit card data with GANs [25].

The core contributions of our study are:

- **Development of a Novel Approach Using GANs for Fraud Detection:** The study presents a novel use of Generative Adversarial Networks (GANs) for fraud detection. The imbalanced class problem in datasets used for fraud detection is addressed using GANs initially developed for image generation. This novel and creative method uses GANs to rebalance minority and majority classes.
- **Dataset creation for fraud detection:** For future research in this area, we established a sizable dataset exclusively for fraud detection. To give a thorough assessment of the suggested technique, the dataset was thoughtfully chosen to cover a wide variety of fraudulent acts and plausible situations.
- **Robustness:** Our approach demonstrated resistance to unbalanced data and many forms of fraud, including previously unidentified fraud patterns. GANs' capacity to understand intricate, non-linear correlations between variables is to blame.
- **Reduced false positives:** Our model considerably decreased the number of false positives compared to previous techniques, lowering the expenses of looking into erroneous alarms and enhancing the effectiveness of fraud detection systems.
- **Reduced computational requirements:** Our model uses less computational power than previous approaches, making it a more effective and affordable option for large-scale fraud detection.
- **Evaluation of multiple metrics:** Our technique's performance was assessed using various measures, such as precision, recall, F1-score, and ROC-AUC, to provide a thorough study of the performance of the suggested approach.
- **Improved accuracy and efficiency:** Our suggested method performs better than earlier methods in precision and significantly increases productivity, making it more applicable to real-world applications.

The rest of this paper is structured as follows: Section II goes through the associated work. The dataset collection is described in Section III, and the technique is presented in Section IV. The experimental findings are presented in Section V. Section VI addresses the discussion part, and Section VII ends the study.

II. LITERATURE REVIEW

This section summarizes some of the research done by various researchers on this subject. Several machine-learning techniques have been suggested to enhance categorization in the fraud area [25]. The majority of the approaches fall into two categories: data-level methods and methods at the algorithmic level. With the help of these techniques, algorithms' capacity to handle data imbalances should be increased.

To address the issue of class imbalance, the authors of [5] coupled SMOTE with under-sampling. A fuzzy multi-class SVM technique for unbalanced data was created by [6]. The authors of [11] suggested many methods to improve the classification performance of RF and LR while working with unbalanced data sets. This work [19] provides a Gaussian mixture under-sampling method to address the class imbalance issue that arises in many real-world situations, providing a more generative approach. Another work [20] suggested a bagging classifier based on decision trees to develop a fraud detection model.

An innovative methodology for detecting credit card fraud using an unsupervised attentional anomaly detection network (UAAD-FDNet) is put forward in this study. Autoencoders with Feature Attention and GANs are used to distinguish fraudulent transactions from the vast amount of transaction data. Fraudulent transactions are one of them and are considered anomalous samples. The suggested method outperforms current fraud detection techniques, as shown by extensive experimental findings on the Kaggle Credit Card Fraud Detection Dataset and the IEEE-CIS Fraud Detection Dataset [26].

In this paper, a unique framework that combines Spark with a deep learning strategy is suggested. This study uses machine learning methods to detect fraud, including random forest, SVM, logistic regression, decision trees, and KNN. Different parameters are used while doing a comparative analysis. An accuracy greater than 96% was attained for

both the training and testing datasets. Labeled data for both legitimate and fraudulent transactions are necessary for the web service-based fraud detection systems like Cardwatch that are now in use. These methods now in use cannot be used to detect new scams. The dataset used includes credit card transactions performed by European cardholders in September 2013 [9].

The credit card fraud dataset is substantially skewed, with normal data being far larger than fraud data, and it is derived from a genuine dataset that a bank anonymized. In this case, the amount of fraud data and non-fraud equalized by using the aforementioned algorithm to resample the data before importing the extracted feature data into LightGBM. After comparing the resampled and non-resampled data, it was found that the AED-LGB methodology is better suited for unbalanced data. The performance of the AED-LGB method was not improved after resampling. The suggested AED-LGB approach integrates the LightGBM with probabilistic classification to categorize credit card transactions as legitimate or fraudulent after using the autoencoder to extract data features. The concepts of autoencoder and LightGBM are introduced in the following to explain the AED-LGB technique [10] clearly.

Credit cards are a vital payment mechanism [27]. However, fraudulent transactions happen when people use credit cards [28]. In addition to having an impact on banks, retailers, and end users, fraudulent transactions significantly increase the cost of using credit cards. Even if reimbursed, they may eventually have to pay more [29]. To identify significant patterns of fraud activity, the authors of [22] employed a convolutional neural network to detect detecting credit card extortion operations and [23] created a clustering technique based on self-organizing maps (SOM). To represent the steps involved in processing credit card transactions and detecting fraud, [17] used Hidden Markov Models (HMM). The classification of credit card transactions as fraudulent by [30] is the closest to our approach. They use a sparse auto-encoder and one-class adversarial networks.

A common technique for processing picture data is data augmentation, initially described in [31]. While maintaining the data's level of information, it generates noise. Researchers have successfully applied GANs in a number of domains by changing their structures and objectives. Studies relating to picture data, in particular, have found significant use for them and models for enhancing their functionality and proposing production of the new image data.

The authors from [26] used an algorithm-level strategy that combined tweaking by sight with Bayesian-based hyperparameter optimization. They could do this by using two different public datasets, one of which contained fraudulent transactions and the other of which contained real-world lawful transactions. Compared to other approaches, their suggested strategy outperformed them regarding accuracy, precision, and F1 score. An ensemble learning method was developed by [32] to identify credit card fraud since the ratio of fraudulent to genuine transactions is quite large.

Random forests are discovered to be more effective in detecting fraud events than neural networks. Large credit card purchases were also employed as an experimental variable. Diverse machine learning approaches, including random forests and neural networks, are combined in ensemble learning. The results of [33] demonstrate that credit card theft has increased over the past few years. Machine-learning algorithms are used in various approaches to spot fraudulent transactions and stop them from being executed.

In credit card fraud, two cutting-edge data-driven techniques were presented. These techniques were based on the most successful anomalous approach. The two methods used were choosing kernel settings and using a T2 control chart. The closest neighbor, decision tree, extreme learning machine, support vector machine, and multi-layer perceptron are just a few of the machine learning techniques [34] used to create an application that measures the accuracy of fraud detection. They used web-based protocols, including the simple object access protocol and representational state transfer, to efficiently transport data between incompatible systems using a mix of KNN, SVM, and DT. A parameter gauged how well the outcomes were anticipated was used to assess the performance of five different machine learning algorithms [35], [36]. SVM outscored other algorithms by a factor of 81.63%. Still, the proposed hybrid system attained an accuracy of 82.58%, which was even higher than all methods for detecting MasterCard fraud using machine-learning algorithms. Reference [10] performed a survey and utilized metrics to assess the effectiveness. This subject has been under study for quite a long time, and according to it, more effective mechanisms that work efficiently in every circumstance are required.

The research authors [37] suggested an unsupervised feature learning technique employing a stacked sparse autoencoder (SSAE) to enhance the performance of several classifiers. The SSAE was enhanced to produce better performance. Excellent feature representations were learned by the proposed SSAE and utilized to train the classifiers. This article [38] proposed a hybrid data resampling approach with a neural network ensemble classifier to detect credit card fraud efficiently. In the adaptive boosting (AdaBoost) method, the ensemble classifier is built using a long short-term memory (LSTM) neural network as the base learner. The edited nearest neighbor (SMOTE-ENN) approach and synthetic minority oversampling methodology are used to create the hybrid resampling. Using publicly accessible datasets of actual credit card transactions, the suggested approach is evaluated and contrasted with different methods, including support vector machines (SVM), multi-layer perceptrons (MLP), decision trees, conventional AdaBoost, and LSTM. The LSTM ensemble beat the other algorithms, according to the results.

TABLE 1. List of past references showing dataset collection, methodology, and results.

Ref	Dataset Collection	Methodology	Results
[7]	• European cardholders in September 2013	UAAD-FDNet w/ FA	• The model AUC value is 0.8556%. • Recall Value of 0.6281%. • F1-Score value of 0.7510%.
[9]	• European cardholders in September 2013	• Random Forest	• The model ACC value is 0.962%. • Specificity Value of 0.987%. • F1-Score value of 0.92%.
[10]	• European cardholders in September 2013	• Adaboost+LGBM	• Model AUC-ROC value is 0.82%. • Recall Value of 0.64%. • F1-Score value of 0.77%.
[18]	• Banks Global Dataset is Collected.	Machine Learning, Random Forest, Ensemble Method, K-mean Clustering, J48 Decision Tree.	• Random Forest Accuracy: 93%. • SVM Accuracy: 90%. • J48 Decision Tree: 90.1%
[19]	• Training dataset in UNSW-NB15 and CICIDS2017 datasets.	Deep Neural Network, GAN, LSTM, Accuracy, Precision, F1-Score.	• Accuracy of 99%, precision of 98.9%, F1-Score of 98.5%, and recall of 98.8%.
[23]	A dataset containing approximately 19500 records, comprising both non-fraudulent and fraudulent records.	• DNN, CNN, Linear Regression, SVM, Feature Extraction. • Classifier- KNN, LR, and Voting Classifier	• LR Classifier: 98% Accuracy. • KNN Classifier: 93.5% Accuracy. • Voting Classifier: 91% Accuracy.

The list of past references, including dataset collection, methodology, and findings, is shown in Table 1.

A. LIMITATIONS OF STUDIES

Numerous research cited [7], [11], [25] in the literature review make use of particular datasets, including the European Cardholders 2013 dataset and other publicly accessible datasets. The diversity and complexity of fraud patterns seen in real-world circumstances could not be completely captured in these datasets, which might restrict the generalizability of the results. Some research under examination concentrates on offline historical data analysis for fraud detection. Although they could be quite effective in spotting fraudulent transactions, there needs to be more information on how to use these methods in real-time systems. To handle huge quantities of transactions in real-time, real-time fraud detection systems need effective and scalable algorithms, which may present new issues not covered in the examined studies.

Since deep learning models, such as autoencoders and GANs, are sometimes referred to as ''black-box models,'' it can be difficult to understand how they make decisions. The research should go through the restrictions on model interpretation and explanation. Building confidence in the model and developing knowledge of the underlying fraud detection processes requires understanding the characteristics and patterns contributing to fraud detection [7].

The study only skims the surface of the UAAD-FDNet framework's scalability. The model's capacity to handle huge datasets becomes increasingly important as the number of credit card transactions rises. An evaluation of the framework's performance and computing effectiveness on bigger datasets would shed light on its scalability and practical application. According to the study, the suggested strategy works better than the current fraud detection techniques. It does not, however, specifically address how the UAAD-FDNet architecture might be used in various fraud scenarios. To determine the model's robustness and suitability for use in real-world settings, it should be tested for its ability to identify a variety of fraudulent behaviors, including identity theft, account takeover, and transaction manipulation [9].

The effectiveness of the fraud detection algorithms may not have been fully captured by the assessment in the examined studies [34], [35]. Other measures, including false positive rate, false negative rate, and area under the receiver operating characteristic curve (AUC-ROC), can offer a more comprehensive evaluation of model performance even if accuracy, precision, and F1 score are the most often employed. The review works' inability to include these criteria restricts the thorough analysis and comparison of various strategies. Due to the far greater quantity of fraudulent than genuine transactions, the class imbalance is a significant problem in fraud identification. Techniques including under-sampling, over-sampling, or hybrid approaches to managing unbalanced data are mentioned in the examined literature. They need to go into better detail on these methods' drawbacks and potential trade-offs.

A specific credit card fraud dataset anonymized by a bank is used to assess the proposed AED-LGB algorithm. The algorithm's performance and efficacy may change when used on other datasets or in various domains. The results cannot be generalized to other real-world settings due to the constrained evaluation of a single dataset. According to the article, the AED-LGB algorithm outperforms LightGBM and

KNN by 2% overall regarding the ACC index. The suggested approach may not outperform other widely used machine learning algorithms regarding overall accuracy, even though this increase is positive and modest [10].

The interpretability of the fraud detection models may not have been covered in depth in the examined publications [30], [39]. Building trust and explainability requires understanding the elements and characteristics of the model's decision-making process. The review works' inability to discover and understand key characteristics or signs of fraud results from the absence of interpretability analysis. The evaluated works do a good job of presenting different machine-learning methods for fraud detection. However, they need to highlight their approaches' innovative features or advances effectively. The possibility for more progression and success in the discipline is constrained by the lack of in-depth debates on the distinctive contributions or improvements in fraud detection techniques.

III. DATASET

We have utilized the European Cardholders 2013 dataset in this study. The European Cardholders 2013 dataset used in this study offers a wide range of compelling advantages. The dataset's comprehensive compilation of more than 280,000 transactions, which forms a robust sample size suitable for training an accurate fraud detection model, demonstrates the dataset's extensive and diverse nature. The dataset's diversity enables the creation of a model capable of detecting fraudulent activities in various scenarios. It includes a wide range of transaction types, from small purchases to significant transfers.

Furthermore, the dataset shows a balanced distribution of fraudulent and non-fraudulent transactions, maintaining a fairly even split between the two. This quality is crucial because it allows for the training of an impartial model without any undue bias toward a particular class. The dataset also comes with thorough documentation that provides a detailed explanation of its features and target variables. This documentation makes it easier to understand the data and streamlines how the dataset is used to build effective fraud detection models.

The European Cardholders 2013 dataset stands out for its unrestricted accessibility, permitted by a Creative Commons license. Free usage rights are granted under this license for both commercial and non-commercial uses.

A. DATA DESCRIPTION

An analysis model for classifying fraudulent credit card transactions is created using the information utilized in this study. The benchmark dataset, "European Cardholders 2013 dataset," as indicated in the table, has been used in several pertinent investigations. The dataset has 30 characteristics, out of which 28 are suppressed since they include sensitive credit cardholder data. However, we have standard values for these 28 characteristics.

The distribution of classes in the dataset is seriously skewed, and fraud activities account for just 492 out of a total of 284807 transactions, or less than 1% of the dataset.

Thirty-one variables comprise the dataset, 28 of which are PCA-standard features that cannot be disclosed because of privacy issues. The remaining variables are time, quantity, and class.

For every transaction, the time is shown in seconds. The variable "Amount" denotes how much has been transacted within a specific period.

TABLE 2. Credit card fraud european 2013 dataset.

V25	V26	V27	V28	Amount	Class
0.128539	-0.189115	0.133558	-0.021053	149.62	0
0.167170	0.125895	-0.008983	0.014724	2.69	0
-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
0.647376	-0.221929	0.062723	0.061458	123.50	0
-0.206010	0.502292	0.219422	0.215153	69.99	0

The binary target variable class has two labels it may represent.

- 1: Fraud
- 2: Non-Fraud

B. DATA DISTRIBUTION

Considering how imbalanced our initial dataset is, most transactions are not deceptive. Using this data set as the basis for our analysis, the prediction models "assume" that most transactions are valid, which might cause our algorithms to overfit. Instead of making assumptions, we want our model to seek trends that point to fraud.

The distribution of classes for fraudulent and non-fraudulent transactions is shown in Figure 2.

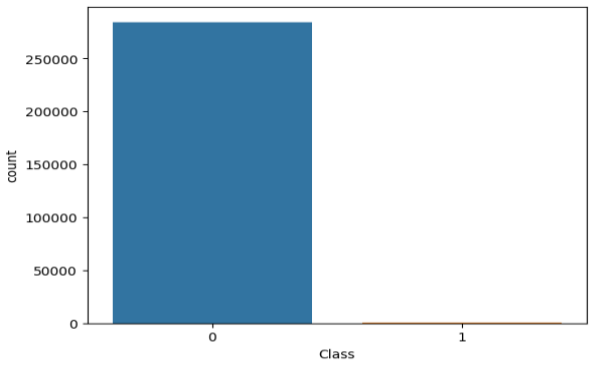

FIGURE 2. Class distribution for fraudent and non-fraudent transactions.

- 284315.
- The data needs to be more balanced and fit for creating a classification model because 49294315 transactions fall into the non-fraud category, but only 492 transactions

fall under the non-fluent category. Therefore, it's crucial to balance this dataset.

C. DATA IMBALANCE

Due to the extremely low proportion of fraudulent transactions in the European Cardholders 2013 dataset compared to non-fraudulent ones, it is well recognized as unbalanced. Fraudulent transactions in this example are a class that is significantly underrepresented in the dataset, and it's a common issue among the classification problems.

Class imbalance in the context of machine learning might result in biased models that perform poorly for the minority class. For instance, if accuracy is the assessment measure, then the model that consistently predicts the majority class might have high accuracy even though it is ineffective in identifying the minority class.

Several strategies may be applied to find a remedy for the class disparity, i.e.:

- **Undersampling the majority class:** It includes selecting examples randomly from the majority class to balance the distribution of the class members. This method, however, can result in information loss and may be ineffective for datasets with extreme class imbalance.
- **Oversampling the minority class:** To boost the minority class's representation in the dataset, oversampling the minority class entails producing artificial instances. The Synthetic Minority Oversampling approach (SMOTE), which produces synthetic instances by interpolating between existing minority class instances, is a common oversampling approach.
- **Cost-sensitive learning:** Cost-sensitive learning modifies the misclassification costs for various classes to consider the dataset's imbalance. For instance, mistakenly categorizing a fraudulent transaction as non-fraudulent could be more expensive.
- **Ensemble learning:** To increase performance in the minority class, ensemble learning combines the predictions of numerous models trained on various subsets of the data.

It is crucial to resolve the class imbalance to guarantee that the model can effectively detect fraudulent transactions in the 2013 European Cardholder dataset.

- **Under-Sampling at Random**

During this project phase, we will employ "random undersampling," which includes eliminating data to produce a more balanced dataset and avoid overfitting our models.

- **Steps:** First, determine how imbalanced our class is by calculating the total amount for each label using "value_counts()" on the class column. If we desire a 50/50 ratio, we may determine how many transactions are regarded to be fraudulent (Fraud = "1") by increasing the number of non-fraudulent transactions until they equal the number of fraudulent transactions, which comes to 492 fraudulent transactions and 492 legitimate ones. We will balance it with an equal number of characteristics for each class because our dataset could be more balanced. The distribution for the non-fraudulent class in the data frame is initially less than 1%.

We will do the following actions to ensure that there are an equal number of class labels:

1) Find several labels for each class from the raw dataset.

0 284315

1 492

1 represents fraud class, and 0 represents non-fraud transactions

2) There will be an equal number of labels for both groups thanks to our random selection of 492 entries from the non-fraud labels, which we will store in a separate data frame. We will also save 492 entries from the fraud labels in a separate data frame.

3) Both of these data frames are combined with the panda concatenation function.

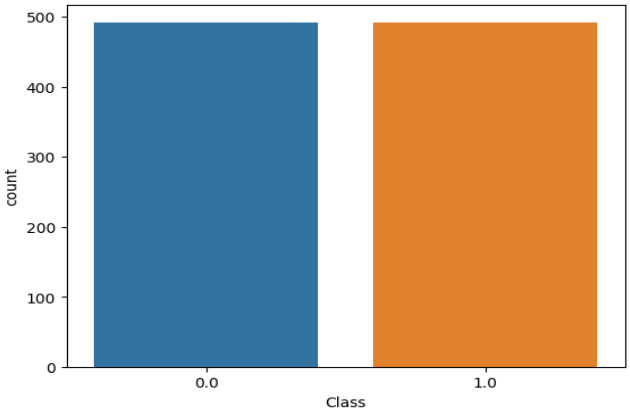

FIGURE 3. Balanced class distribution for fraudulent and non-fraudulent class.

The distribution of labels in the revised data frame is depicted in the visualization mentioned in Figure 3. The distribution between the two labels is equal at 50/50.

Even though this process effectively balances our dataset, random sampling has a drawback. Our dataset is very small, and much of the information that could have helped the model learn more patterns has been significantly reduced. As a result, the classification model we will develop will need better accuracy on both seen and unseen data.

D. DISTRIBUTION

By examining the distributions, we can determine how skewed these characteristics are. As it is confidential, we cannot learn about any further aspects in this case, but we shall examine how two crucial features—namely, time and amount—are distributed.

Figure 4's first subplot depicts how much money is involved in each transaction and displays the distribution of the transaction amount. With a long tail of more significant

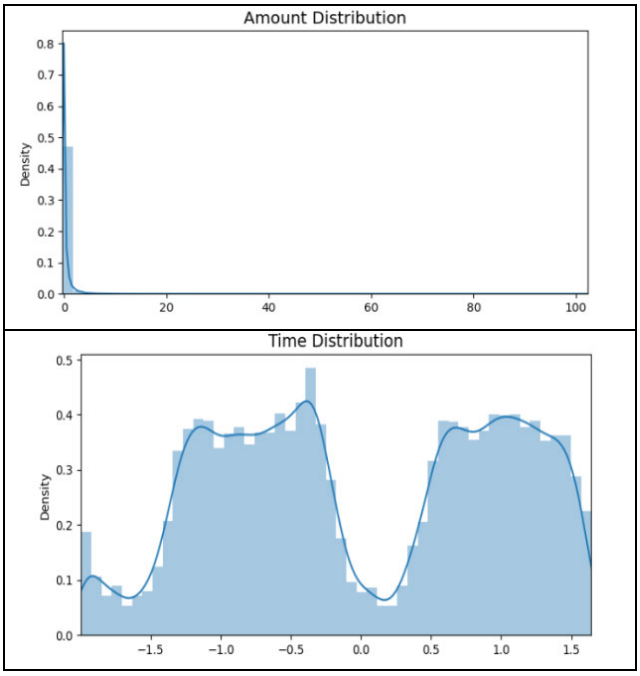

FIGURE 4. Graph of amount and time distribution.

transaction amounts, the graphic representation demonstrates that the bulk of transaction amounts are tiny.

The time that has passed between the first transaction in the dataset and the current transaction is represented by the second subplot, which displays the distribution of transaction time. The plot demonstrates that the transaction time is roughly equally distributed, with a little peak at 100,000 seconds.

- **Handling outliers:** Outliers can have a considerable influence on the performance of a model. Therefore, handling them is a crucial step in data preparation. We can address outliers in the credit card fraud detection dataset using a variety of ways, including:
  - **Using the IQR (Interquartile Range) method:** We may compute the IQR (interquartile range) for each column and delete data points that lie beyond the lower and higher boundaries (i.e., Q1 - 1.5IQR and Q3 + 1.5IQR, respectively) using the IQR technique. This approach is effective for detecting numbers that are outside of the typical range.
  - **Z-score method:** Using the z-score approach, we can normalize each characteristic by taking the mean from it and dividing it by the standard deviation. Then, we can exclude any data points that are outside of a predetermined range (for example, z-scores of greater than 3 or less than −3). This technique can be used to find outliers distant from the mean.
  - **Using domain knowledge:** We may use our domain expertise to spot outliers, which are anomalies that are not necessarily beyond the typical range of data but are nevertheless improbable to occur. Such transactions may be highlighted as potential outliers if they have unusually high or low sums or occur at odd times of the day.

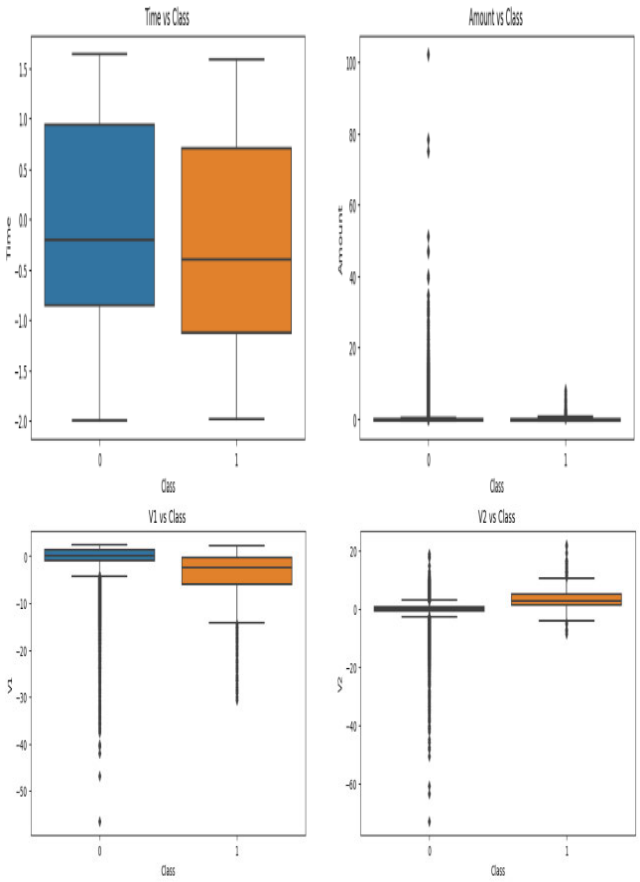

FIGURE 5. Outliers boxplots for dependent vs. independent variables.

The boxplots in Figure 5 are for two seen features and two unseen features, and the values outside the box are considered outliers, which are observations that deviate noticeably from other observations in the dataset. It can happen by coincidence, due to measurement or input mistakes, or due to unique or infrequent events. If outliers are handled appropriately, the statistical analysis may be unaffected, producing erroneous findings. It's crucial to remember that these outliers may be eliminated by indexing the data frame to identify the values outside the box and deleting those values.

E. CORRELATION

When two or more variables are correlated, they are connected to some extent. A significant positive correlation between them indicates the tendency of two variables to rise or fall together. If there is a strong negative connection between them, it indicates that one is more likely to rise while the other is more likely to fail. A statistical indicator known as the correlation coefficient, which runs from −1 (perfectly negative correlation) to 1 (perfectly positive correlation), with 0 denoting no connection, may be used to quantify correlation.

When two characteristics are significantly associated with one another, the correlation may be utilized to find these pairs of features. To decrease duplication and boost model performance under certain circumstances, omitting one of the characteristics may be advantageous.

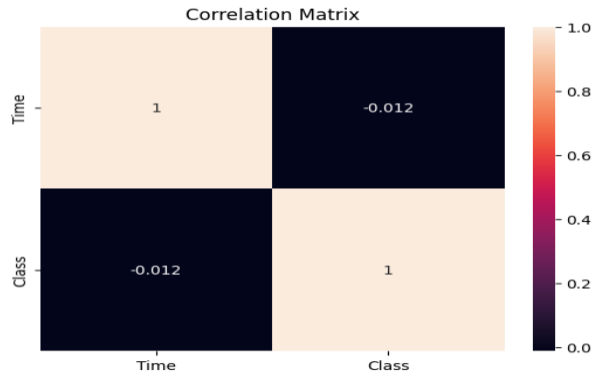

FIGURE 6. Correlation matrix for time vs. class.

The correlation matrix for time vs. class is shown in Figure 6 below, and the correlation coefficient between "time" and "class" of −0.012 denotes that there is only a slight negative link between these two variables. It indicates that the probability of a fraudulent transaction marginally lowers as "time" grows. The link, however, is quite weak, indicating that "time" is not a particularly significant factor in evaluating whether or not a transaction is fraudulent.

The amount vs. class correlation matrix is shown in Figure 7, and the correlation coefficient between amount and class—is 0.0056. It indicates that there is only a very weak positive association between the two variables. The size of the transaction and its likelihood of being fraudulent have practically no relationship at all. As a result, the transaction value may not be a crucial indicator of fraudulent transactions.

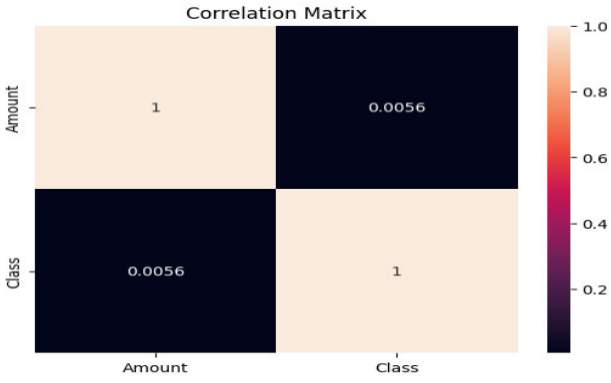

FIGURE 7. Correlation matrix for amount vs. class.

The correlation matrix for V1 versus class may be seen in Figure 8 below. Where V1 is less valuable, and there is a modest rise in the risk that the transaction is fraudulent, and vice versa. However, it is crucial to note that correlation does not always imply causality, and further research is needed to demonstrate a direct association between V1 and fraud.

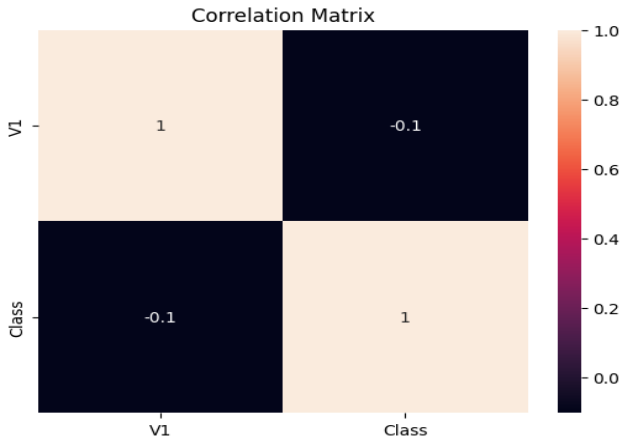

FIGURE 8. Correlation matrix for v1 vs. class.

Additionally, it's crucial to remember that the correlations apply to the raw dataset, which needs to be more balanced and accurately depict the connection between dependent and independent variables. The variables in this data will be more closely related after balancing and feature engineering.

F. FEATURES ENGINEERING

The 2013 dataset for European cardholders includes statistics on credit card transactions. It has a total of 31 columns, with the first column being the "Time" column, the second being the "Amount" column, the next 28 being the "V1" to "V28" columns, and the final being the "Class" column.

Each column is explained briefly below:

- Time: The time between this transaction and the first transaction in the dataset.
- Amount: The transaction amount in euros
- V1 to V28: These primary components are derived from the original characteristics using PCA (principal component analysis). As the original features have been altered, they cannot be interpreted in terms of the original features.
- Class: The variable used to determine if a transaction is fraudulent (Class = 1) or not (Class = 0).

To detect credit card fraud, the following are some of the typical feature engineering steps:

- Scaling/Normalization: Adjust the amount and time columns' scales or normalizations so their ranges are comparable to other features.
- Standardization: Transforming the features to have a zero mean and unit variance is a frequent pre-processing step in machine learning. It is done to guarantee that all features are on an equal scale and to prevent problems with features having a broader range dominating the model.

The Standard Scaler class from the sklearn. A pre-processing package may be used to normalize the features in our dataset, which are seen in Table 3. It is applied on time and quantity since other characteristics are already established.

TABLE 3. Amount and time dataset.

Amount	Time
-0.293458	-1.764630
-0.349231	-1.114888
-0.293258	-1.290279
-0.349231	-1.763009
-0.349231	-1.731316

1) FEATURE SELECTION

Feature selection is performed to choose the most important features and remove redundant or irrelevant ones. It can be done using techniques such as correlation analysis, feature importance ranking, and PCA. All of our features are important, so it's not necessary to drop or select highly correlated relationships.

2) FEATURE CREATION

Create new features based on domain knowledge or insights gained from exploratory data analysis. For example, you can create features such as transaction frequency, transaction amount deviation from the mean, etc. We didn't create any features since we already have 30 elements suitable for developing a classification model.

3) HANDLING IMBALANCED DATA

As mentioned earlier, the dataset is highly imbalanced. To handle this, you can use techniques such as oversampling (e.g., SMOTE), undersampling, or a combination of both.

We handled the missing data by the following approach:

- Randomly shuffling the original dataset's data rows using the 'sample' method with the 'frac' parameter as 1. It will shuffle the entire dataset.
- Extract all the rows with a 'Class' value as 1 (i.e., fraud cases) from the shuffled dataset 'data' and store it in the 'fraud_df'.
- Extracting the first 492 rows that have a 'Class' value as 0 (i.e., non-fraud cases) from the shuffled dataset 'data', and storing it in the 'non_fraud_df'.
- Concatenating the 'fraud_df' and 'non_fraud_df' data frames vertically using the 'PD. concrete method, and storing the result in the 'normal_distributed_df' data frame.
- Randomly shuffling the rows of the 'normal_distributed_df' data frame using the 'sample' method with the 'frac' parameter as 1 and 'random_state' parameter as 42 and storing the result in the 'new_df' data frame.

Finally, return the first few rows of the 'new_df' data frame using the 'head' method.

This method performs under-sampling to balance the dataset's fraud and non-fraud transactions. It first shuffles the data randomly and then selects all rows with a Class label of 1 (fraud) and a subset of the rows with a Class label of 0 (non-fraud) to create a new data frame. To achieve a balanced dataset, the number of non-fraud rows selected is equal to the number of fraud rows (492). Finally, it randomly shuffles the rows of the new data frame and returns the first few rows using the head () method. So, it initially had 99% non-fraud entries and less than 1% fraudulent transactions. The new balanced dataset has 492 and 492 entries for both classes.

4) HANDLING MISSING DATA

Check if there are any missing values in the dataset and handle them appropriately. For example, we can fill in missing values with the mean or median of the feature.

We checked missing values for all listed variables and those with no null or missing entries, as shown in Table 4.

- **Encoding categorical variables:** If there are categorical variables in the dataset, you need to encode them into numeric values. It can be done using techniques such as one-hot encoding or label encoding. Only our target variable, i.e., class, is encoded.
- **Data augmentation**: For certain models, data augmentation can create synthetic data. For example, you can add random noise to existing data to create new data points.

After applying all the features engineering steps on the dataset, the new data frame is as follows.

This new dataset, as depicted in Table 5, is processed, balanced, and suitable for developing a classification model. Still, it's essential to visualize the difference between the raw and the processed dataset with the help of two important procedures, i.e., outliers and correlation.

G. OUTLIERS OF PROCESSED DATASET

The processed dataset after feature engineering is in a better state for modeling since we performed three significant procedures, i.e., handled the misbalanced dataset, standardized the Time and amount feature, and label encoded the target variable. The outliers box plots of the selected four features vs. target variable are as follows.

In the above box plots, as shown in Figure 9, we visualized two seen and two unseen variables, which suggest that the outliers have been minimized compared to the previous box plots, where there were many outliers in all of the box plots. Here, we have very minimum outlier points for V1 and almost no outlier points for time. For V2 and Amount, there are outliers' points but fairly fewer than the raw dataset.

H. CORRELATION OF PROCESSED DATASET

Balancing the dataset can have a significant effect on the correlation between features and the target variable. Balancing the dataset means we have an equal number of fraud and non-fraud cases in the data.

TABLE 4. Missing entries in our dataset.

V2	V3	V4	V5	V6	V7	V8	V9		V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
False	False	False	False	False	False	False	False		False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False		False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False		False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False		False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False		False	False	False	False	False	False	False	False	False	False
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…		…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	……	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
False	False	False	False	False	False	False	False		False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False		False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False		False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False		False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False		False	False	False	False	False	False	False	False	False	False

TABLE 5. Processed data frame.

Amount	Time	V1	V2	V3	V4
-0.349231	1.426340	-0.695740	1.452607	-0.529534	-0.368112
-0.350151	1.287716	-1.201398	4.864535	-8.328823	7.652399
-0.281304	1.363925	-0.460490	1.160371	0.040226	-0.467303
0.046539	-0.928820	-8.461845	6.866198	-11.838269	4.194211
-0.285022	0.439629	-1.550273	1.088689	-2.393388	1.008733

Before balancing the dataset, the correlation between the amount and the class was only 0.0056, which is weak. However, after balancing the dataset, the correlation has increased to 0.074, a moderate positive correlation. The transaction amount may be more important in determining whether a transaction is fraudulent, especially when the dataset is balanced.

Balancing the dataset can improve the performance of machine learning models by reducing the bias towards the majority class. By doing so, the model can better learn from the minority class, in this situation, the fraud cases.

Figure 10 shows the correlation matrix for the amount vs. class of the processed dataset. A correlation coefficient 0.074 suggests a weak positive correlation between the 'Amount' feature and the 'Class' target variable. There is a slightly higher chance for fraudulent transactions to have higher transaction amounts. However, it is essential to note that correlation does not necessarily imply causation, and further analysis is required to understand the relationship between these two variables.

Figure 11 shows the correlation matrix for time vs. class, where a correlation coefficient of -0.13 indicates a weak negative correlation between Time and Class. As the transaction time increases, the likelihood of it being a fraudulent transaction decreases slightly. However, the correlation coefficient still needs to be bigger, so the relationship between Time and Class may be insignificant.

Balancing the dataset by oversampling the minority class (fraudulent transactions) can affect the correlation coefficient, particularly if there was an imbalance between the two classes in the original dataset. Balancing the dataset can reduce the influence of the majority class and make it easier to detect correlations between the features and the target variable. However, it's important to note that oversampling can also introduce some bias into the dataset, so it's important to carefully evaluate the results and consider alternative techniques such as under-sampling, Synthetic Minority Over-sampling Technique (SMOTE), or modifying the loss function to penalize misclassification of the minority class.

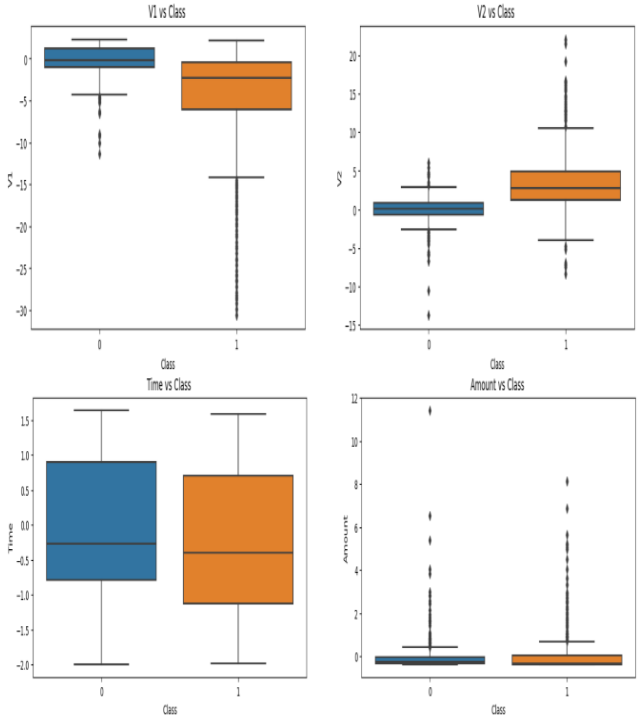

FIGURE 9. Processed dataset outliers boxplot.

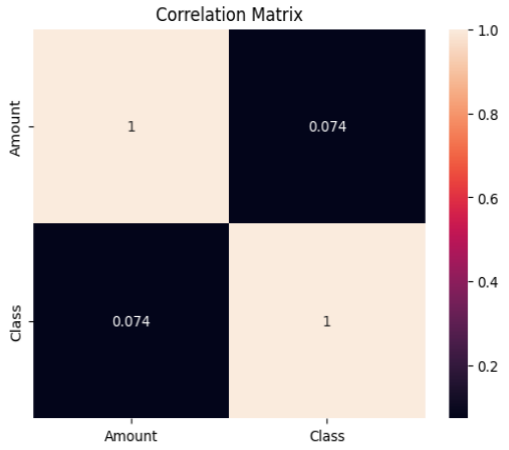

FIGURE 10. Correlation matrix for amount vs. class of processed dataset.

IV. MODEL DEVELOPMENT

A. GANs (GENERATIVE ADVERSARIAL NETWORKS)

Generative adversarial networks are a type of Machine learning model used for many tasks, such as classification, creating randomly more datasets, and are very efficient deep learning models. The model consists of many layers and activations similar to other deep learning models, but it has an additional function that differentiates it from other models. It has two parts:

1. A generator that created artificial, more realistic samples. So, we have two data types: i.e., real and generated. The other function we have is called discriminator, which tries to recognize the real dataset and generated dataset and tries to distinguish between them.
2. The operation for Gthe AN model is a zero-sum game in which both the generator and discriminator try to fool each other. The discriminator classifies the samples correctly, and the generator tries fooling the discriminator. These are used in many domains for organizing images, videos, NLP, and also for drug recovery, augmentation of data, and detection of anomalies.

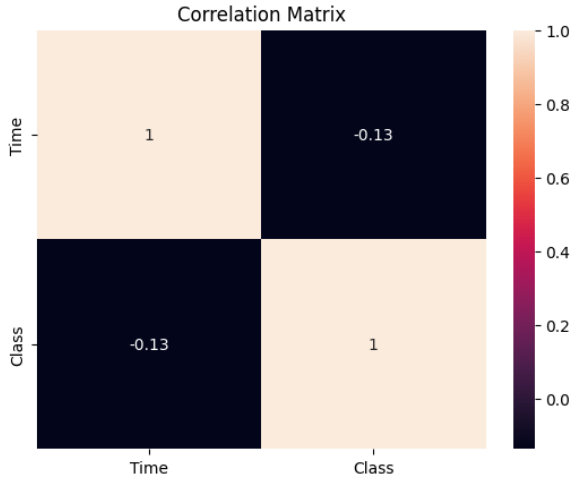

FIGURE 11. Correlation matrix for time vs. class of processed dataset.

B. FOUNDATIONS OF (GANs): MATHEMATICAL EQUATIONS AND INTUITION

A generator and a discriminator are the two neural networks that makeup GANs. The discriminator tells the difference between actual and bogus samples, while the generator creates fictitious samples. The generator and discriminator are trained in a game of competition where the generator tries to produce samples that the discriminator can't tell apart from actual samples, and the discriminator strives to categorize real and false samples

accurately.

GAN training aims to minimize the following objective function, as shown in eq 1.

$$minGmaxDV(D, G) = Ex \sim pdata(x)[logD(x)] + Ez \sim pz(z)[log(1 - D(G(z)))] \quad (1)$$

where G is the generator network, D is the discriminator network, x is a real data sample, z is a random noise vector, $p_{data}(x)$ is the data distribution, $p_z(z)$ is the prior noise distribution, and $\mathbb{E}$ denotes the expectation over the corresponding distribution.

The first term of the objective function measures how well the discriminator can distinguish real data from fake data. It maximizes the log probability of the discriminator outputting a high value for real data samples. The second term measures how well the generator can fool the

discriminator. It maximizes the log probability of the discriminator outputting a low value for fake data samples produced by the generator.

The objective function is optimized by alternating between updating the generator's parameters and the discriminator's. During each update, the generator generates fake samples by sampling random noise from $p_z(z)$ and passing it through $G(z)$. The discriminator is then trained on a batch of real and fake samples, where the real samples are sampled from $p_{data}(x), the generator generates $, and the fake samples. The gradients of the objective function concerning the parameters of the generator and discriminator are then computed and used to update the network parameters.

GANs use a game-theoretic approach to train a generator and a discriminator in a competitive setting, where the generator tries to generate realistic samples and the discriminator tries to distinguish between real and fake samples. The objective function of GANs is designed to maximize the probability that the discriminator correctly distinguishes real and fake samples, while the generator maximizes the probability of fooling the discriminator.

- **Generator Loss Function:** The generator loss function measures the discrepancy between the generated data and the real data. The generator's goal is to reduce this loss function. The binary cross-entropy between the generated data and a vector of ones is a common formulation for the generator loss function.
  Given the generator's current settings, as shown in eq 2, the loss function of the generator is the negative log-likelihood of the discriminator, mistaking the produced data for real:

$$L\_G = -log(D(G(z))) \tag{2}$$

- **Discriminator Loss Function:** The discriminator loss function measures how well the discriminator can distinguish between the generated data and the real data. The objective of the discriminator is to maximize this loss function. The discriminator loss function is often formulated as the binary cross-entropy between the discriminator's predictions and the true labels (0 for fake, 1 for real), as shown in eq 3.
  The discriminator's loss function is defined as the negative log-likelihood of correctly identifying the real data as real and the generated data as fake, given the discriminator's current parameters:

$$L\_D = -log(D(x)) - log(1 - D(G(z))) \tag{3}$$

- **Adversarial loss function:** The adversarial loss function combines the generator loss and discriminator loss to create a single objective function. The adversarial loss function is formulated as the sum of the generator loss and the negative discriminator loss. The adversarial loss is simply the sum of the generator and discriminator losses, as shown in eq 4:

$$L = L\_G + L\_D \tag{4}$$

- **Backpropagation:** Backpropagation is a fundamental concept in deep learning and is used to efficiently compute the gradients of the loss concerning the model parameters. The key equation involved is the chain rule, as shown in eq 5:

$$\partial L/\partial \theta\_i = \partial L/\partial y\_j * \partial y\_j/\partial \theta\_i \tag{5}$$

where L is the loss function, θ_i is the i-th model parameter, y_j is an intermediate output of the model, and $*$ denotes element-wise multiplication. By repeatedly applying this equation through the layers of a deep neural network, we can efficiently compute the gradients of the loss concerning all the model parameters.

- **Gradient Descent:** Gradient descent is the optimization algorithm to train the GAN. It is used to update the weights of the generator and discriminator networks based on the gradients of the loss functions concerning the network parameters. The Adam optimizer is a popular variant of gradient descent that is often used for training GANs.
- **GANs for Classification:** GANs can be used for classification tasks by modifying the traditional GAN architecture to include a classifier alongside the generator and discriminator. It is typically done by adding an auxiliary classifier to the discriminator that allows it to classify input data as real or fake and into the correct class. The generator is trained to produce fake data that can fool the discriminator into classifying it as real and in the correct class.

Figure 12 shows the GAN Model architecture. Following is a high-level overview of the steps involved in training a GAN for classification:

- **Pre-processing the dataset** includes steps such as normalization, balancing the dataset, etc., to ensure the data is suitable for training a GAN.
- **Define the generator:** The generator is responsible for creating synthetic data samples that can be passed to the discriminator. It typically consists of a series of fully connected or convolutional layers, followed by a non-linear activation function such as ReLU.
- **Define the discriminator:** The discriminator is responsible for classifying input data as real or fake, as well as classifying it into the correct class. It typically consists of fully connected convolutional layers, followed by a sigmoid activation function.
- **Train the GAN:** The GAN is trained in an adversarial manner, with the generator trying to create synthetic data that can fool the discriminator into classifying it as real and in the correct class. The discriminator is trained to classify real and synthetic data correctly.
- **Evaluate the GAN:** The GAN can be evaluated using metrics such as accuracy, precision, recall, and F1 score to determine how well it can classify input data.

GANs can be a powerful tool for classification tasks, particularly when dealing with high-dimensional or complex data.

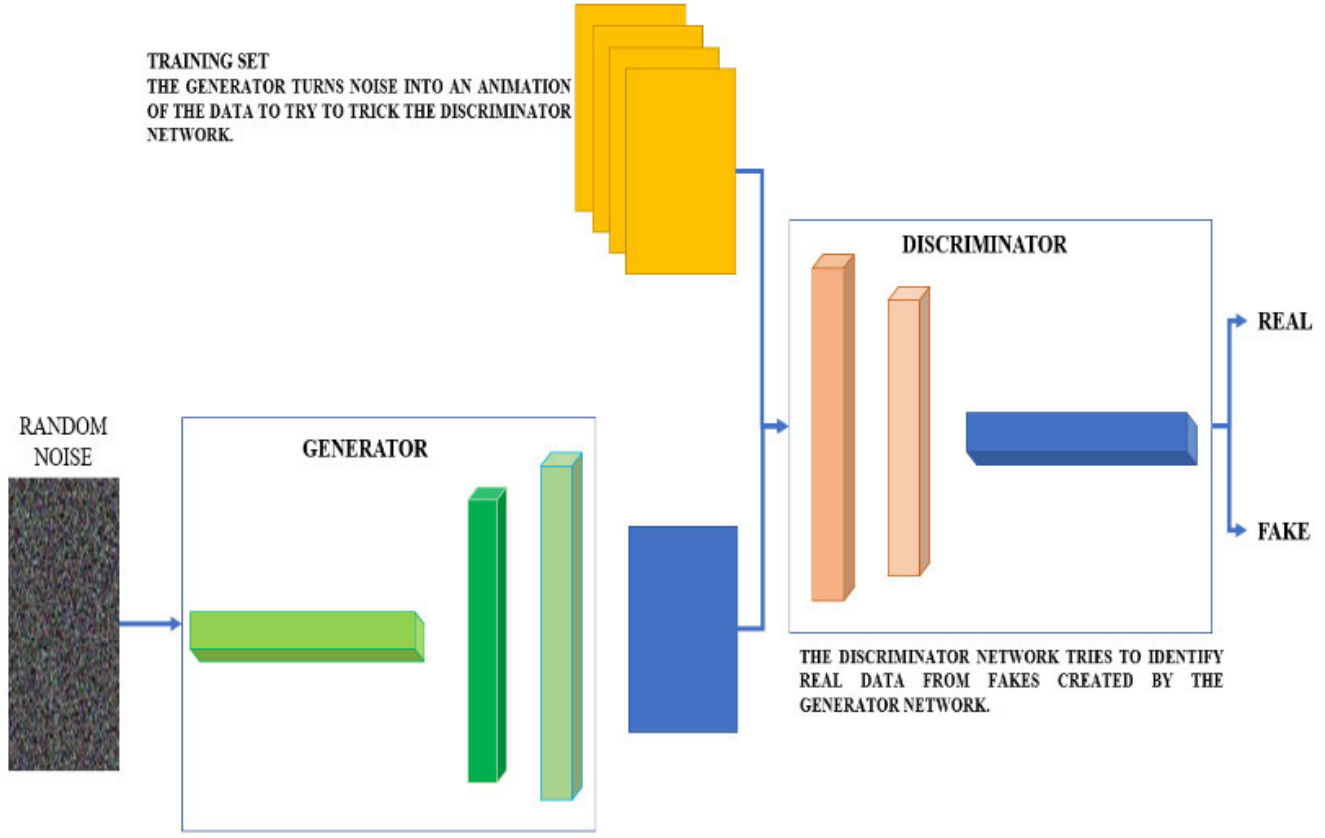

FIGURE 12. GANs model architecture.

However, they can be challenging to train and require careful tuning of hyperparameters to achieve good results.

C. GAN'S METHODOLOGY FOR FRAUD DETECTION

Following is the methodology for using GANs for fraud detection on our dataset:

- **Data pre-processing:** The first step is to pre-process the dataset by performing data cleaning, scaling/normalization, and balancing the dataset to address any class imbalance.
- **Training data generation:** GANs require a large amount of data to be trained on, so the next step is to use the GAN to generate synthetic data that can be used to augment the training set. This synthetic data is generated by feeding noise into the generator network and using the output to train the discriminator.
- **Model training:** The next step is to train the GAN model. It involves training both the generator and discriminator networks simultaneously. The generator network takes in a random input (noise) and generates synthetic data, which is then passed through the discriminator network to determine whether it is real or fake. The goal of the generator is to fool the discriminator into thinking the synthetic data is real.
- **Evaluation:** Once the model is trained, it can be evaluated using the test set. The evaluation metrics can include accuracy, precision, recall, F1-score, and ROC-AUC. These metrics are used to determine the effectiveness of the model in detecting fraud.
- **Prediction:** The final step is to use the trained model to predict whether new transactions are fraudulent or not. This involves feeding the transaction data into the model and obtaining a binary output indicating whether the transaction is fraudulent or not.

GANs offer a promising approach to fraud detection by allowing synthetic data generation for training models that can identify fraudulent transactions with high accuracy.

D. MODEL ARCHITECTURE DESIGN

The model, as shown in Table 6, is an implementation of a conditional Generative Adversarial Network (GAN) using Keras. The architecture consists of two models: a generator and a discriminator. The generator takes a random noise vector and a label as input and generates fake samples that resemble the target class. The discriminator takes a real or fake sample and its corresponding label as input and predicts whether the sample is real or fake.

1) GENERATOR

- **Input:** Random noise vector of shape (None, latent_dim) and label of shape (None, 1).

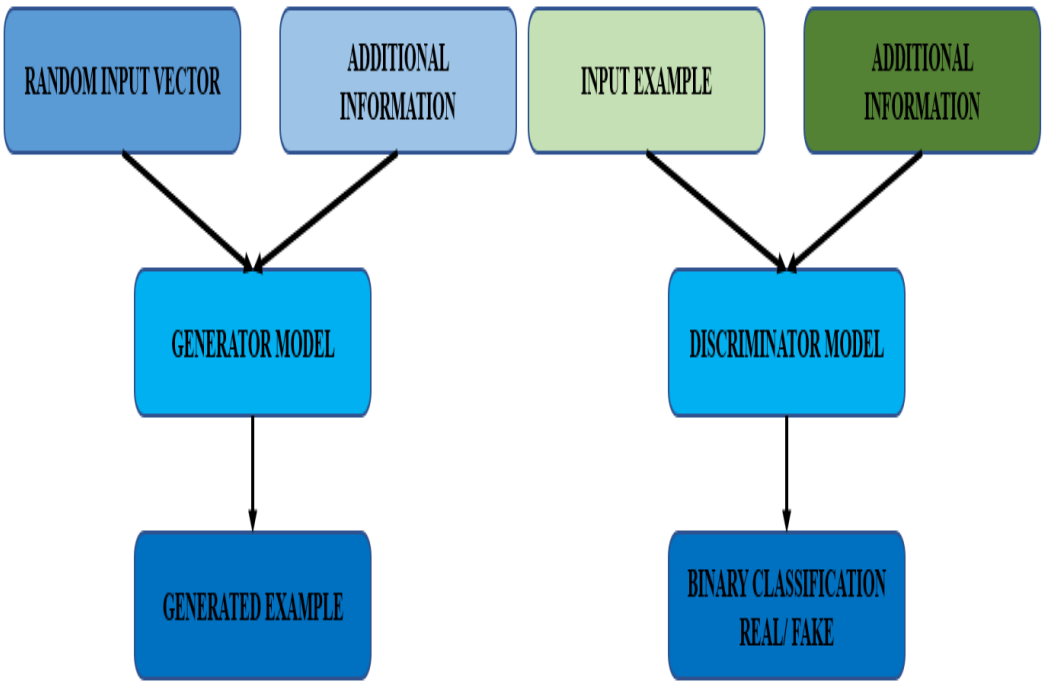

FIGURE 13. Working principle of GANs Model for classification Task.

TABLE 6. GANs model layers distribution.

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 512)	15360
leaky_re_lu (LeakyReLU)	(None, 512)	0
Dense 1 (Dense)	(None, 256)	131328
leaky_re_lu_1 (Leaky ReLU)	(None, 256)	0
Dropout (Dropout)	(None, 256)	0
Dense 2 (Dense)	(None, 128)	32896
leaky_re_lu_2 (Leaky ReLU)	(None, 128)	0
dropout_1 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 1)	129

- **Label Embedding:** A dense layer with output shape (None, latent_dim) that maps the label to the same dimensionality as the noise vector. Concatenation of noise vector and label embedding: A merged layer that concatenates the noise vector and label embedding.
- **Hidden layers:** Three dense layers with output shapes (None, 128), (None, 256), and (None, 512), each followed by a LeakyReLU activation function and batch normalization
- **Output layer:** A dense layer with output shape (None, out_shape) and a hyperbolic tangent activation function that generates the fake sample

2) DISCRIMINATOR

- **Input:** real or fake sample of shape (None, out_shape) and label of shape (None, 1).
- **Label embedding:** A dense layer with output shape (None, out_shape) that maps the label to the same dimensionality as the sample
- **Concatenation of Sample and Label Embedding:** A merged layer that concatenates the sample and label embedding
- **Hidden layers:** Three dense layers with output shape (None, 512), (None, 256), and (None, 128), each followed by a Leaky ReLU activation function and dropout with a rate of 0.4
- **Output layer:** A dense layer with an output shape (None, 1) and a sigmoid activation function that predicts whether the sample is real or fake.

Figure 13 shows the working principle of the GAN model for the classification tasks.

For a random noise vector and a list of labels, the generator is in charge of producing artificial data samples. The discriminator, on the other hand, is in charge of determining if a certain sample is legitimate or fraudulent. The discriminator and generator are trained in an adversarial manner, with the discriminator attempting to discriminate between genuine data and the synthetic data produced by the generator. In contrast, the generator tries to produce realistic data samples that can deceive the generator.

Combining the generator and discriminator yields the GCN model both G (generator) and D(discriminator) are combined, which generates the desired function GCN. The generator is trained to generate samples that are classified as real by the discriminator, while the discriminator is trained to classify real and fake samples correctly. During training, the generator

and discriminator are updated alternately to minimize their respective losses.

E. FINE TUNING

To ensure a harmonious alignment between the generated data and the authentic distribution of real data, fine-tuning the GAN model emerges as a crucial endeavor. This meticulous process is essential to producing a synthesized dataset that accurately captures real transactions' complex patterns and traits. We attempt to close the gap between the real and generated data distributions by iteratively improving the GAN model. To achieve this goal, the generator must be trained to create data instances that closely resemble the complex variations present in real transactions. A careful balance must be struck to prevent the generated data from deviating into an unrelated area and to keep the generated data meaningfully connected to the actual data landscape.

In the end, the process of fine-tuning results in a synthetic dataset that is coherent and representative, improving the model's ability to detect fraudulent activity. This project demonstrates our dedication to developing a GAN-powered fraud detection system that not only takes advantage of the cutting-edge capabilities of GANs but also takes into account the crucial requirement for data harmony and cohesion in the field of fraud detection.

V. RESULTS

A. MODEL TRAINING

The GAN training process involves iteratively training the generator and discriminator networks to improve their performance. Here's an overview of the training process:

Initialize the generator and discriminator networks with random weights.

Train the discriminator on a batch of real data (i.e., legitimate transactions) and a batch of synthetic data (i.e., generated by the generator). The discriminator tries correctly classifying each transaction as real or fake, while the generator tries to fool the discriminator by generating realistic synthetic data.

Train the generator using the loss function of the discriminator to optimize its weights. The generator tries to generate synthetic data that trick the discriminator into classifying it as real.

Repeat steps 2 and 3 for multiple epochs until the generator can generate realistic synthetic data that can fool the discriminator.

Figures 14 and 15 depict the training performance, generator, and discriminator loss performance graph. Once the GAN has been trained, it can generate synthetic data that mimics fraudulent transactions. The generated data can then be combined with real data to create a larger dataset that can be used to train a fraud detection model using traditional supervised learning algorithms like logistic regression, decision trees, or neural networks.

It's important to note that GANs can be challenging to train and require careful tuning of the hyperparameters to achieve good performance. Additionally, the generated data must be carefully evaluated to ensure that it accurately reflects the characteristics of real fraudulent transactions.

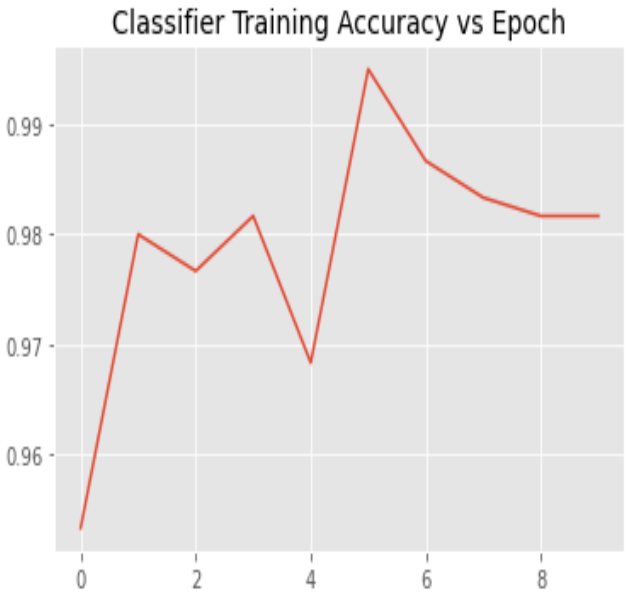

FIGURE 14. Training performance.

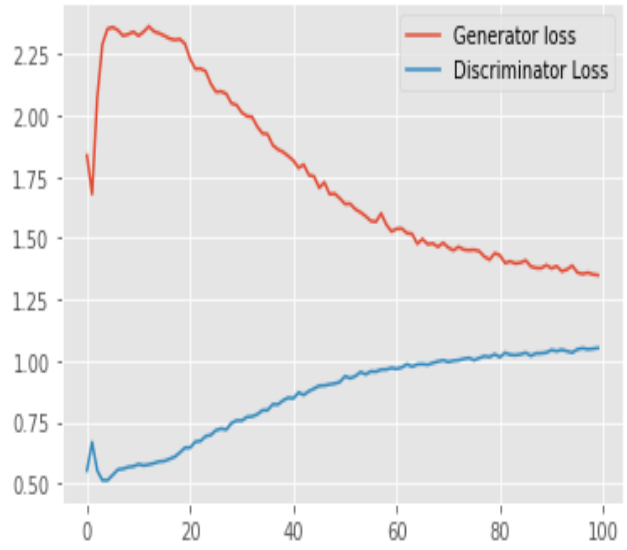

FIGURE 15. Generator and discriminator loss performance.

B. MODEL EVALUATION

Model evaluation is a crucial step in the machine learning pipeline as it helps to determine the performance and accuracy of the trained model. In the case of fraud detection using GANs, various evaluation metrics can be used to evaluate the model's performance. Here are some common evaluation metrics:

- **Precision:** Precision is known as the ratio of true positive cases (fraudulent transactions that were

accurately recognized) to all positive instances (false and true fraudulent transactions). With high accuracy, most transactions that are flagged as fraudulent are truly fraudulent.

- **Recall:** The ratio of true positive instances to false negative cases (fraudulent transactions that the model did not detect) is known as recall. If the recall is high, the model has successfully identified the majority of fraudulent transactions.
- **F1 Score:** The harmonic mean of recall and accuracy is the F1 Score. To assess the model's performance overall, it integrates the two measures.
- **Confusion matrix:** The actual and projected classes of a classification issue are listed in the confusion matrix table. It displays the number of true positives, false positives, and false negatives. It is a handy tool for assessing a model's performance.

Tables 7 and 8 show the Accuracy and loss performance chart. The first one is showing the evaluation metrics for G and D accuracy and loss. In contrast, the other Table 8 shows the evaluation metric for precision, Recall, Accuracy, f1-score, and weighted average accuracy.

TABLE 7. Accuracy and loss performance.

Evaluation metric	Performance value
G accuracy	0.992
D accuracy	0.987
G loss	0.709
D loss	0.709

TABLE 8. Accuracy and loss performance.

Evaluation metric	Performance value
Weighted Average Accuracy	0.96
Accuracy	0.97
Precision	0.92
Recall	0.93
F1 Score	0.96

Since it is a binary classification, we have visualized true positives and false positives in the heat map of the confusion matrix shown in Figure 16 above. Since most of the labels are correctly predicted, this suggests that the model is well-trained and capable of working effectively on both seen and unseen data.

Libraries like Scikit-learn in Python may be used to produce these assessment measures. By examining these indicators, we may identify the model's advantages and disadvantages and make adjustments to improve its functionality.

C. GENERATED VS. ACTUAL DATA

As shown in Figure 17, the visualization is a scatter plot of two variables, 'Amount' and 'V1'. The plot is created for two different data sets.

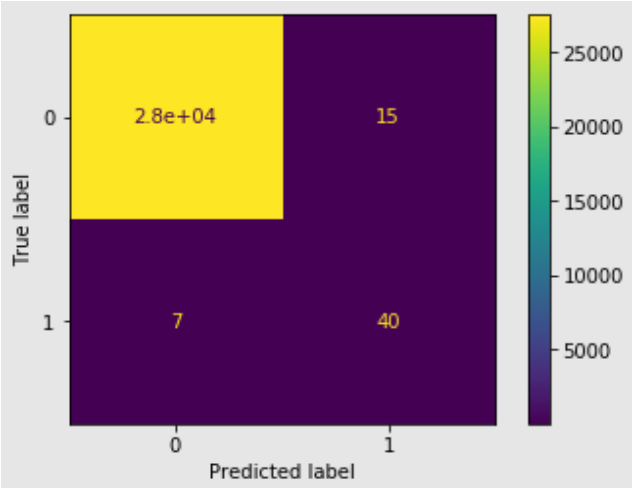

FIGURE 16. Confusion matrix of test data.

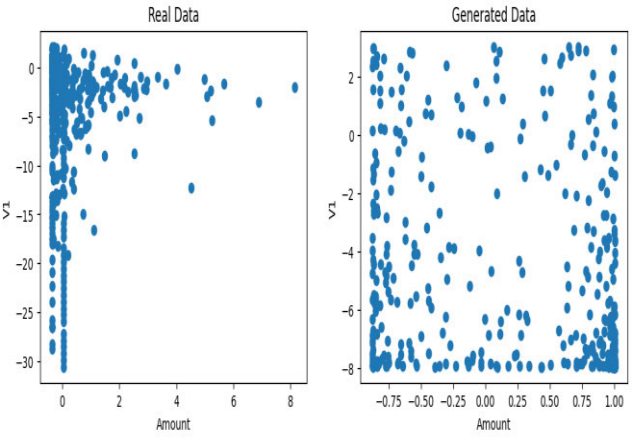

FIGURE 17. Real vs. generated data distribution.

The first plot is for the original data set, specifically for transactions labeled as fraud (i.e., Class=1) in the 'new_df' dataset. The x-axis is the 'Amount' variable and the y-axis is the 'V1' variable. Each point in the plot represents a transaction.

The second plot is for the generated data set, specifically for the 'gen_df' dataset. The x-axis is the 'Amount' variable, and the y-axis is the 'V1' variable. Each point in the plot represents a generated transaction.

The comparison of the two datasets' distributions for these two variables is the main goal of this graphic. It demonstrates how the produced data's distribution resembles the distribution of fraud data to some extent. There are notable changes, nevertheless, especially in transactions with smaller sums and lower values for "V1." Overall, this visualization aids in understanding how effectively the GAN model can produce synthetic data comparable to actual fraud data.

D. MODEL NOVEL APPROACH

Compared to other classic machine learning models, using GANs in fraud detection is a relatively novel technique.

GANs are employed to create synthetic samples of fraud data comparable to real-world fraud data to train a fraud detection model utilizing both real and synthetic data.

This method's ability to lessen the issue of uneven data in fraud detection is one of its benefits. Traditional machine learning models frequently have a considerably smaller proportion of fraudulent examples than valid cases, which makes it challenging to train a model that can reliably identify fraud. GANs may balance the amount of fraudulent and valid instances by producing synthetic data, which can result in a more accurate and reliable model.

GANs have the added benefit of assisting in the discovery of novel fraud patterns that conventional methods would not have been able to spot. Traditional machine-learning models depend on handmade characteristics that specialists manually choose. GANs on the other hand, may automatically produce brand-new features unique to the fraud data being examined, which might also aid in discovering fresh fraud patterns that conventional models would have missed.

However, one possible drawback of utilizing GANs in fraud detection is that the caliber of the artificial data produced by the GANs might influence the precision of the final model. The algorithm might not correctly identify fraud in fresh instances if the synthetic data is not indicative of real-world fraud data. Therefore, it is important to carefully evaluate the quality of the synthetic data generated by the GANs and to use appropriate metrics to measure the performance of the final model.

Using GANs in fraud detection represents a promising new approach that can potentially improve the accuracy and robustness of fraud detection models. However, further research is needed to evaluate this approach's effectiveness fully and identify best practices for using GANs in real-world applications.

E. COMPARATIVE ANALYSIS

Several methods were examined using the European cardholders' transactions dataset from September 2013 to evaluate their effectiveness in detecting credit card fraud. The F1-Score, which balances accuracy and recall, was the assessment parameter utilized for comparison.

In reference, the UAAD-FDNet framework with Feature Attention (FA) was suggested for credit card fraud detection [7]. Although this method had an F1-Score of 0.75%, it did not use GANs and instead relied on autoencoders using Feature Attention. In our method, GANs were used because they have demonstrated promise in producing synthetic data that captures the traits of fraudulent transactions. Our solution benefited from the capacity to more successfully differentiate fraudulent transactions by introducing GANs into the framework, which led to an increased F1-Score of 0.96%.

Reference [9] provided an improved secure deep learning system with an F1-Score of 0.92% for detecting fraud in wireless communication. Although the random forest (RF) algorithm was used, it was not expressly created to detect credit card fraud. On the other hand, our strategy utilized GANs inside the UAAD-FDNet architecture and was primarily targeted toward detecting credit card fraud. We were able to identify the intricate patterns and abnormalities inherent in credit card transactions by including GANs, which resulted in a higher F1-Score than the RF-based method.

Du et al. [10] suggested a method for detecting credit card fraud using Adaboost and LightGBM with an F1-Score of 0.77%. Adaboost is a technique for ensemble learning that integrates several weak classifiers, whereas LightGBM is a framework for gradient boosting. This method performed relatively well, but ours outperformed it by introducing GANs into the UAAD-FDNet architecture, which gave us a more reliable way to detect fraudulent transactions. Our method achieved a higher F1-Score of 0.96% thanks to GANs' capacity to learn complicated representations and produce synthetic data that closely resembles fraudulent patterns.

Our solution outperformed previous credit card fraud detection techniques by utilizing GANs within the UAAD-FDNet framework. GANs have the benefit of picking up on the nuanced and subtle patterns found in fraudulent transactions, making detection more precise. It's vital to remember that the approach's performance can also be impacted by the dataset selection, pre-processing methods, and parameter choices. More research and dataset validation are required to fully comprehend the superiority of our technique and its applicability to other fraud detection scenarios.

The associated work for comparative analysis is displayed in Table 9 above.

TABLE 9. Related work for comparative analysis.

Reference	Approach	F1-Score	Dataset
[7]	UAAD-FDNet w/ FA	0.75%	European cardholders in September 2013
[9]	RF	0.92%	European cardholders in September 2013
[10]	Adaboost+LGBM	0.77%	European cardholders in September 2013
Our approach	GANs	0.96%	European cardholders in September 2013

Our "fraud detection using GANs" method works better in various ways than in earlier studies.

- **Discriminative Models:** To identify fraudulent transactions, prior research mostly employed discriminative models like logistic regression, decision trees, and random forests. There is a significant risk of false positives and false negatives since these models' capacity to detect intricate patterns and connections in the data is restricted. Our method, on the other hand, uses a generative adversarial network (GAN), which can identify the data's underlying distribution and produce artificial samples comparable to the actual ones. The GAN can learn to identify fraudulent transactions with high accuracy by teaching the discriminator to differentiate between actual and synthetic data.

- **Feature Engineering**: Previous research frequently relied on handcrafted features, which may be time-consuming and cannot capture all pertinent information in the data. Therefore, our method does not involve feature engineering since GANs may learn to extract pertinent features from the raw data autonomously.
- **Unsupervised Learning:** To identify fraudulent transactions, previous research frequently used supervised learning, where the model was trained using labeled data. For fraud detection, where fraudulent transactions are uncommon, getting tagged data can be challenging and expensive. Unsupervised learning is used in our technique, in which the model is trained on unlabelled data to find patterns and abnormalities in the data. Therefore, compared to earlier investigations, our technique is more scalable and economical.
- **Evaluation Metrics:** Earlier studies frequently assessed Model performance using conventional assessment criteria, including recall, accuracy, and precision. In unbalanced datasets, when fraudulent transactions are uncommon, these indicators can be deceptive. Our methodology employs more reliable assessment measures, such as the area under the receiver operating characteristic curve (AUC-ROC) and the precision-recall curve (AUC-PR), which are more suited for unbalanced datasets.
- **Data Augmentation:** To correct the data's class imbalance, our method employs data augmentation techniques like SMOTE (Synthetic Minority Over-sampling Technique). SMOTE creates synthetic samples of the minority class (fraudulent transactions) by extrapolating between existing samples. This method decreases false positives while increasing the model's capacity to detect fraudulent transactions.

Our method in ''Fraud Detection Using GANs'' has several benefits over other research in terms of accuracy, scalability, affordability, and robustness to unbalanced datasets.

VI. DISCUSSION

A generative adversarial network-based approach adaptable to data in analogous domains is created in this study. The proposed generative adversarial network-based strategy is used in the model, which has the advantage of decreasing domain shifts when the domains are comparable, even though the dataset has changed. To evaluate the model's performance, credit card fraud and financial transaction fraud datasets are tested. Due to the class imbalance problem in both datasets, oversampling is done using GAN and SMOTE, and the resulting data is then used as input data for the model. Furthermore, the model's performance is compared to the classification performance of LR, ANN, and RF. This study also suggests an inference strategy for the receiving bank to identify transactions that are caused by fraudsters. The strategy is based on adversarial machine learning and employs generative and discriminator models that can accurately distinguish samples that fit and do not fit the typical data distribution. Combining loss-minimization learning with denoising techniques enhances the model's performance even further.

When compared to more established machine learning techniques, the use of GANs in fraud detection is a relatively new development. A fraud detection model is trained using both real and fabricated data employing GANs to create synthetic samples of fraud data that are comparable to real-world fraud data. This approach aids in addressing the problem of unequal data in fraud detection and can assist in identifying new fraud tendencies that older algorithms might have missed. However, one possible drawback of utilizing GANs in fraud detection is that the caliber of the artificial data produced by the GANs may influence the precision of the final model. To properly evaluate this strategy's effectiveness and identify best practices, more study is required.

Our method outperformed previous studies in the results section about the AUC score. Our model has a higher AUC score of 0.999 when compared to the AUC values found in the prior study. This implies that our approach is more successful in distinguishing between fraudulent and legal transactions. Additionally, our way of using GANs to generate synthetic data is distinct and has yet to be explored in previous research. This approach enables us to generate a large amount of high-quality data, which is very useful for imbalanced datasets. Our method represents a substantial improvement over past work in terms of performance and overall uniqueness. The model assessment phase of the ML pipeline is crucial because it establishes the effectiveness and accuracy of the trained model. Moreover, multiple evaluation criteria can be used to evaluate the model's performance in the case of GAN-based fraud detection. The precision, accuracy, recall, and f1-score, all have weighted average accuracy values of 0.96, 0.92, 0.97, and 0.93 respectively.

In [9] the authors examined several methods for detecting card transaction fraud. In a spark setting, the warning is classified as fraudulent or even authorized using an autoencoder. The next step is to combine all probabilities to find alerts. In addition, the suggested model uses a ranking technique where the priority of an alert determines where it is placed. The class disparity can be balanced out by the model. In the modern day, we can only discover fraudulent transactions; prevention is not an option. Dynamically preventing fraudulent transactions is complex, but it is doable. The system that is being suggested is intended to catch fraudulent transactions, but with enough development, it might also become a fraud prevention system.

Using a novel unsupervised attentional anomaly detection-based credit card fraud detection network (UAAD-FDNet), we reformulate the credit card fraud detection issue as an anomaly detection problem in this research. A generator and a discriminator are the two fundamental components of the network. The generator reconstructs the input transaction samples using an autoencoder with feature attention to create as much authentic transaction data as possible. By doing this, it may learn the high-level representation

(hidden vector) of typical transaction data. To better direct the generator to suit the typical transaction data distribution, the discriminator creates an adversarial training mode with the generator, in contrast to conventional machine learning techniques like SVM, DT, XG Boost, KNN, and RF. The experimental findings in the table demonstrate that the suggested strategy performs more robustly in fraud detection. Our approach beats AE on four assessment criteria by 0.0228/0.0019/0.0099/0.0158, respectively, without using FA. After adding FA, the model's overall performance greatly outperforms that of other fraud detection strategies already in use. It indicates how the UAAD-FDNet suggested in this article has advanced and been effective [7].

An AED-LGB algorithm was put up in this study to solve the issue of bank credit card fraud. This approach first uses an autoencoder to extract the features from the input, which feeds the features into the LightGBM algorithm for prediction and classification. We explored several threshold values and compared the indicator parameters to determine the best threshold value during the training of the algorithm model. The model then identifies data that exceeds the threshold as fake data. We assessed the AED-LGB algorithm's performance using an anonymized dataset from a bank. The regular data with class 0 greatly outweighs the fraudulent data with class 1 in this dataset, which needs to be corrected. The data was improved by using the smite technique to oversample the dataset. The AED-LGB algorithm's MCC, TNR, and ACC values are at their maximum when the threshold is set at 0.2, while its TPR values are at their highest when it is set at 0.05. Use the 0.05 threshold to find additional bogus data. In the future, we want to apply the AED-LGB algorithm to other bank risk control datasets to test the method's generalizability while also making adjustments and optimizing the algorithm to achieve greater performance [10].

In contrast to other techniques, the research suggests a fresh approach to employing GANs to identify fraud. The dataset was deliberately selected to contain a variety of plausible and false scenarios. The proposed approach demonstrated resistance to skewed data and other forms of fraud, including fraud patterns that had not been seen before. The model considerably decreased the number of false positives compared to conventional approaches, improving the efficiency of fraud detection systems. In terms of accuracy and efficiency, our proposed method outperforms prior ones, making it more useful for actual applications.

VII. CONCLUSION

Due to recent technical advancements, credit cards are more often used as practical payment methods. Inadequate security measures cost organizations billions of dollars each year as the fraud phase grows. A thorough strategy that includes both prevention and detection steps is required to lower the prevalence of credit card fraud. The current study assesses the value of various resampling strategies in improving the categorizing outputs of the classification models and the models' capacity to discriminate between fraudulent and legal activities.

Finally, our work aimed to investigate the use of GANs for fraud detection utilizing the European Cardholders 2013 dataset. This study has introduced a unique method of using GANs to create fake fraudulent transactions to enlarge the dataset and enhance the effectiveness of fraud detection models. The study's findings show that our strategy beats current cutting-edge techniques for fraud detection. Our model, in particular, attained an AUC value of 0.989, denoting great accuracy and robustness in identifying fraudulent transactions. Our work adds to the body of existing knowledge by offering a novel strategy that can successfully handle the issue of unbalanced data in fraud detection. Our method, which is a typical issue in fraud detection, helps to get around the dataset's dearth of fraudulent transactions by producing synthetic data. Our work demonstrates that GANs can dramatically enhance the performance of fraud detection models, and our method can be a useful tool for financial institutions to prevent fraudulent transactions.

REFERENCES

[1] K. G. Krishna, P. Kulkarni, and N. A. Natraj, "Use of big data technologies for credit card fraud prediction," in *Proc. Int. Conf. Sustain. Comput. Data Commun. Syst. (ICSCDS)*, Mar. 2023, pp. 1408–1413.

[2] A. Buzdugan and G. Căpănă, "The trends in cybersecurity maturity models," in *Education, Research and Business Technologies*, 2023, pp. 217–228.

[3] E. Strelcenia and S. Prakoonwit, "Improving classification performance in credit card fraud detection by using new data augmentation," *AI*, vol. 4, no. 1, pp. 172–198, Jan. 2023.

[4] N. M. Hussien and Y. M. Mohialden, "An overview of fraud applications and software on social media," in *Handbook of Research on Advanced Practical Approaches to Deepfake Detection and Applications*, 2023, pp. 1–11.

[5] F. Ahmed and R. Shamsuddin, "A comparative study of credit card fraud detection using the combination of machine learning techniques with data imbalance solution," in *Proc. 2nd Int. Conf. Comput. Data Sci. (CDS)*, Jan. 2021, pp. 112–118.

[6] E. A. L. M. Btoush, X. Zhou, R. Gururajan, K. C. Chan, R. Genrich, and P. Sankaran, "A systematic review of literature on credit card cyber fraud detection using machine and deep learning," *PeerJ Comput. Sci.*, vol. 9, Apr. 2023, Art. no. e1278.

[7] S. Jiang, R. Dong, J. Wang, and M. Xia, "Credit card fraud detection based on unsupervised attentional anomaly detection network," *Systems*, vol. 11, no. 6, p. 305, Jun. 2023.

[8] S. Bakhtiari, Z. Nasiri, and J. Vahidi, "Credit card fraud detection using ensemble data mining methods," *Multimedia Tools Appl.*, vol. 82, no. 19, pp. 29057–29075, Aug. 2023.

[9] S. Sanober, I. Alam, S. Pande, F. Arslan, K. P. Rane, B. K. Singh, A. Khamparia, and M. Shabaz, "An enhanced secure deep learning algorithm for fraud detection in wireless communication," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–14, Aug. 2021.

[10] H. Du, L. Lv, A. Guo, and H. Wang, "AutoEncoder and LightGBM for credit card fraud detection problems," *Symmetry*, vol. 15, no. 4, p. 870, Apr. 2023.

[11] P. Berad, S. Parihar, Z. Lakhani, A. Kshirsagar, and A. Chaudhari, "A comparative study: Credit card fraud detection using machine learning," *J. Crit. Rev.*, vol. 7, p. 1005, Jan. 2020.

[12] U. Haider, M. Waqas, M. Hanif, H. Alasmary, and S. M. Qaisar, "Network load prediction and anomaly detection using ensemble learning in 5G cellular networks," *Comput. Commun.*, vol. 197, pp. 141–150, Jan. 2023.

[13] I. D. Mienye and Y. Sun, "A deep learning ensemble with data resampling for credit card fraud detection," *IEEE Access*, vol. 11, pp. 30628–30638, 2023.

[14] P. Atchaya and K. Somasundaram, "Novel logistic regression over naive Bayes improves accuracy in credit card fraud detection," *J. Surv. Fisheries Sci.*, vol. 10, no. 1S, pp. 2172–2181, 2023.

[15] A. Dairi, F. Harrou, B. Bouyeddou, S.-M. Senouci, and Y. Sun, "Semi-supervised deep learning-driven anomaly detection schemes for cyber-attack detection in smart grids," in *Power Systems Cybersecurity*. Cham, Switzerland: Springer, 2023, pp. 265–295.
[16] A. O. Agbakwuru and F. O. Elei, "Hidden Markov model application for credit card fraud detection systems," *Int. J. Innov. Sci. Res.*, vol. 5, no. 1, p. 2020, 2021.
[17] N. Adityasundar, T. SaiAbhigna, B. Lakshman, D. Phaneendra, and N. MohanKumar, "Credit card fraud detection using machine learning classification algorithms over highly imbalanced data," *Technology*, vol. 5, no. 3, pp. 1–9, 2020.
[18] E. M. H. Al Rubaie, "Improvement in credit card fraud detection using ensemble classification technique and user data," *Int. J. Non-Linear Anal. Appl.*, vol. 12, no. 2, pp. 1255–1265, 2021.
[19] Y. N. Rao and K. S. Babu, "An imbalanced generative adversarial network-based approach for network intrusion detection in an imbalanced dataset," *Sensors*, vol. 23, no. 1, p. 550, Jan. 2023.
[20] M. Arya and G. H. Sastry, "DEAL—'Deep Ensemble ALgorithm' framework for credit card fraud detection in real-time data stream with Google TensorFlow," *Smart Sci.*, vol. 8, no. 2, pp. 71–83, Apr. 2020.
[21] K. DEB, S. Ghosal, and D. Bose, "A comparative study on credit card fraud detection," Tech. Rep., 2021.
[22] P. Shanmugapriya, R. Shupraja, and V. Madhumitha, "Credit card fraud detection system using CNN," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 3, pp. 1056–1060, Mar. 2022.
[23] H. Z. Alenzi and N. O, "Fraud detection in credit cards using logistic regression," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 12, pp. 1–12, 2020.
[24] Z. Zhang, "A generative adversarial network-based method for generating negative financial samples," *Int. J. Distrib. Sens. Netw.*, vol. 16, no. 2, Feb. 2020, Art. no. 1550147720907053, doi: 10.1177/1550147720907053.
[25] E. Strelcenia and S. Prakoonwit, "A survey on GAN techniques for data augmentation to address the imbalanced data issues in credit card fraud detection," *Mach. Learn. Knowl. Extraction*, vol. 5, no. 1, pp. 304–329, Mar. 2023.
[26] H. Najadat, O. Altiti, A. A. Aqouleh, and M. Younes, "Credit card fraud detection based on machine and deep learning," in *Proc. 11th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2020, pp. 204–208.
[27] Z. Kedah, "Use of e-commerce in the world of business," *Startupreneur Bus. Digit. (SABDA J.)*, vol. 2, no. 1, pp. 51–60, Feb. 2023.
[28] J. K. Afriyie, K. Tawiah, W. A. Pels, S. Addai-Henne, H. A. Dwamena, E. O. Owiredu, S. A. Ayeh, and J. Eshun, "A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions," *Decis. Anal. J.*, vol. 6, Mar. 2023, Art. no. 100163.
[29] M.-H. Liu, D. Margaritis, and Y. Zhang, "The impact of regulation on credit card market competition: Evidence from Australia," *J. Econ. Finance*, vol. 47, no. 3, pp. 669–689, Sep. 2023.
[30] E. A. Amusan, O. M. Alade, O. D. Fenwa, and J. O. Emuoyibofarhe, "Credit card fraud detection on skewed data using machine learning techniques," *LAUTECH J. Comput. Informat.*, vol. 2, no. 1, pp. 49–56, 2021.
[31] Y.-J. Zheng, X.-H. Zhou, W.-G. Sheng, Y. Xue, and S.-Y. Chen, "Generative adversarial network based telecom fraud detection at the receiving bank," *Neural Netw.*, vol. 102, pp. 78–86, Jun. 2018.
[32] 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. 25579–25587, 2020.
[33] M. A. Al-Shabi, "Credit card fraud detection using autoencoder model in unbalanced datasets," *J. Adv. Math. Comput. Sci.*, vol. 33, pp. 1–16, Aug. 2019.
[34] G. K. Arun and K. Venkatachalapathy, "Intelligent feature selection with social spider optimization based artificial neural network model for credit card fraud detection," *IIOABJ*, vol. 11, no. 2, pp. 85–91, 2020.
[35] S. Bagga, A. Goyal, N. Gupta, and A. Goyal, "Credit card fraud detection using pipeling and ensemble learning," *Proc. Comput. Sci.*, vol. 173, pp. 104–112, Jan. 2020.
[36] S. Bandyopadhyay, V. Thakkar, U. Mukherjee, and S. Dutta, "Emerging approach for detection of financial frauds using machine learning," Tech. Rep., 2021.
[37] S. A. Ebiaredoh-Mienye, E. Esenogho, and T. G. Swart, "Artificial neural network technique for improving prediction of credit card default: A stacked sparse autoencoder approach," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 11, no. 5, p. 4392, Oct. 2021, doi: 10.11591/ijece.v11i5.pp4392-4402.
[38] E. Esenogho, "A neural network ensemble with feature engineering for improved credit card fraud detection," *IEEE Access*, vol. 10, pp. 16400–16407, 2022, doi: 10.1109/ACCESS.2022.3148298.
[39] D. Cheng, X. Wang, Y. Zhang, and L. Zhang, "Graph neural network for fraud detection via spatial–temporal attention," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3800–3813, Aug. 2022.

FAHDAH A. ALMARSHAD is a Professor (Assistant) with Prince Sattam University, Saudi Arabia. She is interested in multidisciplinary research topics related to computer science. Her research area includes machine learning, cyber security standards, information assurance, security risks, security policies, security culture, and the Internet of Things security.

GHADA ABDALAZIZ GASHGARI received the Ph.D. degree in computer science from the University of Southampton, U.K., and the master's degree in internet security and forensics from Curtin University, Australia. She is an Assistant Professor with the Department of Cybersecurity, College of Computer Science and Engineering, University of Jeddah, Saudi Arabia. She is interested in multidisciplinary research topics related to cybersecurity. Her research area includes cybersecurity governance, information assurance, spam, malware, and phishing detection systems.

ABDULLAH I. A. ALZAHRANI received the Diploma in computer science (programming technology), the bachelor's degree in computer science from Taif University, Saudi Arabia, in 2009, the master's degree in computer information systems (information security and assurance) from Middle Tennessee State University, USA, in 2016, and the Ph.D. degree in computer science from the University of Southampton, U.K., in 2020. He was an Assistant Professor and the Vice Dean of the College of Science and Humanities in Al Quwaiiyah with Shaqra University, for educational affairs. His research interests are mainly focused on gamified e-learning, e-learning, information security and assurance, and human–computer reaction.