Credit Card Fraud Detection using GAN and Feature Engineering

Afeef Ashraf C
Mar Athanasius College Of Engineering Kothamangalam
Ajaz Ali
Mar Athanasius College Of Engineering Kothamangalam
Anand D
ananddivakaran1206@gmail.com
Mar Athanasius College Of Engineering Kothamangalam
Shabiya M I
Mar Athanasius College Of Engineering Kothamangalam
Rini T Paul
Mar Athanasius College Of Engineering Kothamangalam

Research Article

Keywords: Generative Adversarial Network(GAN), Credit card Fraud Detection, Feature Selection, Feature Engineering, Visual analytics, Data analytics

Posted Date: May 20th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4380806/v1

Additional Declarations: No competing interests reported.

Credit Card Fraud Detection Using GAN and Feature Engineering

Afeef Ashraf C[1*], Ajaz Ali[2*], *Anand D[3*], Shabiya M I[4*] Rini T Paul[5*]
*Department of Computer Science and Engineering
Mar Athanasius College of Engineering, Kothamangalam, Kerala,India
[1]afeefashr030@gmail.com, [2]ajazali7051@gmail.com, [3]ananddivakaran1206@gmail.com,
[4]shabiyaismayil@gmail.com [5]rinitpaul@mace.ac.in

***Abstract*—Credit card fraud is a prevalent financial crime that causes substantial losses to individuals and financial institutions. Traditional fraud detection methods often rely on rule-based systems or supervised machine learning algorithms, which may not be effective in detecting novel or evolving fraudulent patterns. This project proposes a novel approach for credit card fraud detection using a generative adversarial network (GAN) to generate synthetic fraudulent transactions, feature engineering techniques to extract relevant features from the transaction data, and anomaly detection methods to identify fraudulent transactions based on their deviation from normal patterns. The proposed approach involves three main phases: data preparation and feature engineering, GAN-based data balancing, and anomaly detection. Experimental results demonstrate that the proposed approach outperforms traditional fraud detection methods, achieving higher accuracy, precision, recall, and F1-score in identifying fraudulent transactions. Additionally, the approach is more robust to changes in the underlying data distribution and can effectively detect novel or evolving fraudulent patterns.**

***Index Terms*—Generative Adversarial Network(GAN),Credit card Fraud Detection, Feature Selection,Feature Engineering,Visual analytics,Data analytics**

I. Introduction

CYBER fraud involving credit cards is a significant problem for the banking sector, resulting in the loss of billions of dollars yearly.Enhancing cyber security is a task that the banking industry must undertake. Numerous technologies have been developed for the purpose of identifying and monitoring credit card fraud. But because threats are ever-evolving, the banking industry needs to be prepared with the most advanced and effective cyberfraud management systems.

The incidence of credit card cybercrime has increased in recent years due to the surge in the use of credit cards and other online payment methods. Credit card fraud comes in a variety of forms. For example, a credit card may be lost or stolen, or credit card information may be taken through cybercrime.

Moreover, entering personal information when making an online transaction raises the possibility of fraud. Still, researchers in machine learning (ML) have shown interest in the challenging task of credit card cyber theft detection. Credit card-related datasets exhibit a noticeable degree of skewness. Many algorithms struggle to discern between objects belonging to minority classes when working with datasets that exhibit a considerable skew. For cyber fraud detection systems to be effective, they need to react more quickly. Another major area of concern is the effect of novel assault strategies on the conditional distribution of the data over the period.

Generic adversarial networks, or GANs, were developed by [1] and are among the most well-known generating methodologies. The versatility and scalability of GANs to generate fictitious samples for credit card fraud detection are investigated in this paper. This paper seeks to contribute to the existing body of knowledge on GANs for data augmentation in unbalanced classes. Researchers argue that using GANs is the most appropriate and effective option when it comes to machine learning strategies for tackling unbalanced class situations.a sensible and effective tactic in contrast to other machine learning techniques. Its adaptability and capacity to understand underlying data structures make it very resistant to overfitting and overlapping. In order to accomplish this, we examined past studies conducted by experts specialising in fraud detection who used GANs to improve credit card data [2].

Especially when it comes to credit card transaction analysis, feature engineering [3]is vital to the efficacy of fraud detection systems. When working with datasets balanced by Generative Adversarial Networks (GANs), where precise identification of fraudulent activity is crucial, the technique becomes even more crucial. Carefully designed features enable the system to identify faint patterns that point to fraud, improving overall dependability and performance.

Examining transaction metadata to extract important information such transaction amounts, timestamps, and frequencies is one aspect of feature engineering[3]. By providing insightful information about transaction patterns, these metrics help the system identify abnormalities that differ from normal behavior. Furthermore, a fuller picture of account activity is provided by historical behavior analysis, which includes average transaction amounts and habitual transaction durations. This makes it easier to spot inconsistencies that might indicate fraudulent activity.

II. METHODOLOGY

We start our credit card fraud investigation with the raw data. This large dataset is carefully cleaned and preprocessed.

It is distinguished by a notable disparity between authentic and fraudulent transactions. Outliers are subdued, missing values are corrected, and the data is prepared for additional research. Normalisation or transformations may be used to make sure that the next step works. Next, by utilising the capabilities of GANs (Generative Adversarial Networks). We train a generator to produce artificial copies of such elusive fraudulent transactions by carefully selecting an architecture such as WGAN or CGAN. In the end, this competitive process between the discriminating network and the generator produces high-quality synthetic data, which effectively rectifies the imbalance between the classes and gives our models a more balanced representation.

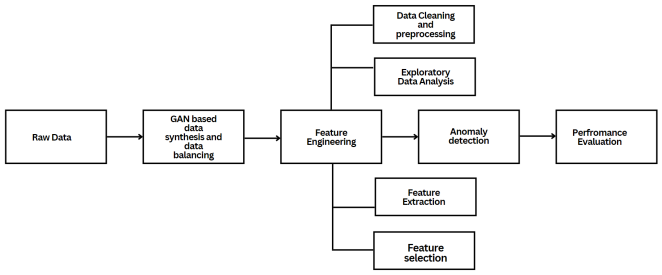

Fig. 1. Work-flow diagram(Block Diagram)

A. GAN-Based Data Balancing

In numerous real-world datasets, particularly those pertaining to classification issues, the amount of data points in one class is frequently noticeably greater than that in another. Machine learning models may encounter difficulties as a result of this class imbalance, which could result in inaccurate predictions and subpar performance. Data balancing is a set of techniques aimed at artificially equalizing the representation of different classes in a dataset. This can be achieved through various methods.

A major difficulty for machine learning models in credit card fraud detection is data imbalance, when the number of valid transactions far outnumbers the number of fraudulent ones. Here's where GANs, or Generative Adversarial Networks, become an effective data balancing tool.[1]

GANs consists of two other neural networks namely the Generator and Discriminator. Generator undergoes rigorous training to produce synthetic fraudulent transactions following the similar patter as seen in the real dataset,Discriminator on the other hand simultaneously be functioning as a rival network honing its ability to discern real fraudulent transactions from the deceptive imitations crafted by the generator.

This creates an adversarial rivalry in which the discriminator ceaselessly improves its detection skills and the generator constantly works to improve the art of forging, creating ever-more-realistic synthetic fraud.By means of this iterative process, the generator eventually develops into an expert forger, with the ability to produce superior synthetic fraudulent transactions. These fraudulent transactions can be strategically combined with the original dataset to provide a more equitable representation. GANs are mainly focused and are mostly used for image generation ,However over the year a number of methods were developed for text-generation and tabular generation also. Our focus is thus on such methods developed and how to refine such technique in favour of our need.

1) Tabular GANs: They bring up a number of issues, which is why creating tabular data presents its own set of difficulties: the multiple data types (int, decimals, categories, time, text), the various distribution shapes (multi-modal, longtail, Non-Gaussian, etc.), the sparse one-hot-encoded vectors, and the severely unbalanced categorical columns.[4]
Let us say table T contains nc continuous variables and nd discrete(categorical) variables and each row is C vector.These variables have an unknown joint distribution P.Each row is independently sampled from P.The object is to train a generative model M.M should generate new a synthetic table Tsynth with the distribution similar to P.A machine learning model learned on Tsynth should achieve a similar accuracy on a real test table T test,as would a model trained on T.

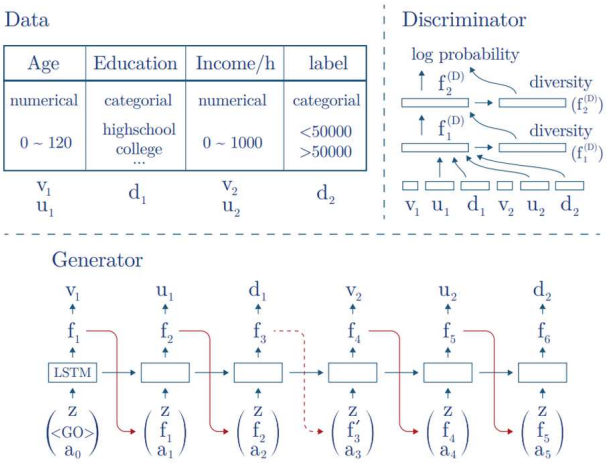

Fig. 2. Example of using TGAN to generate a simple census table. The generator generates T features one be one. The discriminator concatenates all features together. Then it uses Multi-Layer Perceptron (MLP) with LeakyReLU to distinguish real and fake data.[4]

2) CTGAN: CTGAN (Conditional Tabular Generative Adversarial Networks) is considered a better option than TGAN due to its key improvements over previous methods.[5]

In order to handle imbalanced discrete columns, CTGAN utilises training-by-sampling, a conditional generator, and mode-specific normalisation to overcome non-Gaussian and multimodal distribution.[4] This enables CTGAN to produce synthetic data with a distribution resembling the real data and to evenly explore all possible values in discrete columns. CTGAN performs better since it is also trained using WGAN loss with gradient penalty.

Synthetic rows conditioned on a discrete column can be produced by CTGAN's conditional generator. CTGAN can efficiently explore all possible discrete values by sampling the conditional vector and training data according on the log-frequency of each category. In this case, this allows us to

synthesize more fraud transactions by supplying precise conditions. This method improves the performance of producing tabular data by resolving the issues of unequal data distribution and unbalanced categorical columns.
To summarise, the primary distinctions between TGAN and CTGAN are centred around how they provide extra data, how they train boosting in an adversarial fashion, and the particular stages that are involved in the modelling and assessment procedures. In contrast to TGAN, CTGAN adds training-by-sampling, a conditional generator, and mode-specific normalisation to accommodate imbalanced discrete columns. These variations aid in CTGAN's enhanced ability to provide synthetic tabular data.

3) SDV:Synthetic Data Generation Library in Python: The Python SDV module[6] is a useful tool for creating synthetic data. It may be applied to a variety of data-related activities and offers an adaptable and efficient solution in situations where genuine data might not be easily accessible.
It offers a variety of tools for generating synthetic datasets, which can be especially helpful in situations where access to real data is restricted or if using real data is prohibited due to privacy concerns. Users can create synthetic data with the SDV library for a range of uses, such as data analysis, machine learning, and privacy protection.
SDV library can be used to generate synthetic fraud transaction by providing the appropriate constraints and conditions and can sample out the required data instances ,it has means to do so.

B. Feature Engineering

In the context of the credit card fraud detection project, feature engineering is the process of turning unstructured transaction data into useful characteristics that let machine learning algorithms tell the difference between authentic and fraudulent transactions. This procedure includes taking transaction amounts, merchant categories, transaction kinds, and timestamps out of the dataset and extracting them as numerical, categorical, and temporal properties. Feature engineering helps identify potential fraudulent activities by identifying important patterns and correlations within the data based on unique behavioural patterns and abnormalities.

Additionally, feature engineering helps the project by resolving problems with data quality and making it easier to create new features. Imputation, outlier detection, and normalisation are a few techniques that help make sure the dataset is reliable and appropriate for modelling. Through improving the calibre and variety of features,feature engineering enables machine learning models to capture complex relationships and make accurate predictions regarding the likelihood of fraud. Additionally, the interpretability of the models is improved as feature engineering highlights the most influential factors contributing to fraud detection, providing valuable insights for stakeholders in devising effective risk management strategies and fraud prevention measures.

A visual analytics tool called FeatureEnVi [7] was created expressly to help with machine learning (ML) feature engineering. Feature engineering is the process of turning unprocessed data into features that improve machine learning models' ability to predict the future. The inadequate feature engineering support in the current visual analytics tools for machine learning is the goal of FeatureEnVi. Users can experiment with various feature generating combinations, select the most crucial characteristics, and convert initial features into potent substitutes. In addition to providing users with a visual guide and statistical data regarding the relevance of each item, the system offers numerous automatic feature selection strategies. Furthermore, by leveraging data space slicing, FeatureEnVi allows users to investigate the effects of features at both local and global scales.

When it comes to fraud detection, FeatureEnVi can be used to find key features that greatly enhance the prediction ability of machine learning models that are employed in fraud detection systems. With the help of FeatureEnVi, users may choose, modify, and create features, which can enhance the precision and effectiveness of fraud detection algorithms. In fraud detection, where the identification of pertinent features is essential for accurate detection of fraudulent activities, the system's ability to visualise the impact of features and assist users in the feature engineering process can improve the transparency and trustworthiness of the process. All things considered, FeatureEnVi offers a thorough visual analytics solution for feature engineering, which is advantageous when it comes to machine learning applications like fraud detection.

C. Model selection and Prediction

The goal of a machine learning project's model selection and prediction phase is to find the best model for the job at hand by carefully analysing and contrasting various techniques. In this process, a variety of algorithms are usually taken into consideration, each with unique properties and applicability for particular data types and issue areas. The selection process depends on variables like model interpretability, computational efficiency, and predictive performance. Conventional techniques like logistic regression and decision trees are among the simpler approaches; more sophisticated techniques like support vector machines and neural networks are used. Extensive assessment criteria are utilised to objectively evaluate the models' performance, such as accuracy, precision, recall, and F1-score. To guarantee robustness and generalisation over various datasets, cross-validation techniques are frequently added.

Alongside model evaluation, hyperparameter adjustment is essential to improving the predictive power of the chosen models. In order to maximise model performance and generalisation capacity, hyperparameters are methodically changed using techniques like as grid search, random search, or Bayesian optimisation. Through this iterative process, models are better able to catch subtle patterns and nuances within the data, improving their predicted accuracy on samples that have not yet been seen. Finally, the results of painstaking model selection and hyperparameter tuning lead to the identification of the best machine learning model for the given task, providing decision-makers with strong predictive insights and opening up

applications in a variety of fields, including marketing, finance, healthcare, and more.

III. Proposed Design

A. Data Collection and Preprocessing

For this research, we used the European Cardholders 2013 dataset. The European Cardholders 2013 dataset, which is what this study used, has many strong benefits. The vast and diverse nature of the dataset is demonstrated by its thorough compilation of over 280,000 transactions, which creates a robust sample size appropriate for training an accurate fraud detection algorithm. The richness of the dataset makes it possible to develop a model that can identify fraudulent activity in a range of contexts. It covers a broad spectrum of transaction kinds, from little buys to substantial transfers. Since the dataset had previously undergone PCA transformation, the majority of data instances are vector values that correspond to various numerical, category, and other values,Hence no preprocessing of the dataset was required.

B. Data Synthesis

1) TabGAN approach: GAN will be employed in the generation of the synthetic fraud data in order to balance.A Python package called TabGAN[4] is intended to be used with Generative Adversarial Networks (GANs) to produce tabular data. It focuses on tabular datasets in particular, which are different from the image data that is typically utilised in GAN applications. Created by Diyago Silva, it provides features like adversarial filtering and sampling for data production, making it possible to perform jobs like balancing uneven datasets and augmenting data. The dataset was synthesised by tabgan , However there were no provision to change or alter the data distribution in the generated samples , Hence the generated ones also conatined the similar data distribution when tested with the random forest classifier the dataset increased the precsion of the model.

2) SDV approach: The acronym for Synthetic Data Vault is SDV[6]. This Python package is intended to create artificially generated versions of datasets with structure. Using a variety of methods, SDV is able to replicate the fundamental dependencies and structure of the original data, producing realistic synthetic data samples that maintain statistical attributes like correlations, distributions, and feature-to-feature interactions.
The Synthetic Data Vault (SDV) library contains a special model called CTGAN[5]. Conditional Tabular Generative Adversarial Network is referred to as CTGAN. It's a generative model made especially to produce artificial tabular data. Using the power of Generative Adversarial Networks (GANs), CTGAN uses the original tabular dataset to understand the underlying data distribution. It then creates synthetic samples that closely mimic the original data while maintaining its statistical integrity.

Essentially, SDV offers a framework for creating synthetic data, and among the models accessible inside SDV, CTGAN is one that is especially designed to produce synthetic tabular data. These technologies come be handy for a number of uses, including model testing, data augmentation, and privacy-preserving data sharing in situations where real data access is constrained. Without jeopardising the confidentiality or privacy of sensitive data in the original dataset, they allow users to create synthetic datasets that may be used for testing and training machine learning models. SDV gives the synthesiser the ability to apply external conditions and, in turn, receive instructions on the properties of the data instances that are being created.

C. Feature Engineering

1) Feature Selection: Feature selection is a foundational step wherein a diverse range of methods, including filter, wrapper, embedded, or hybrid approaches[8], are employed to discern the importance of different features. By visualizing the outcomes of these techniques, insights can be gleaned into their efficacy and their impact on model performance. Feature transformation is another crucial aspect, where alternative transformations are applied based on statistical measures such as target correlation, mutual information, and per class correlation. Feedback derived from these measures guides the selection of appropriate transformations, ensuring they align with the data's characteristics and modeling objectives.

In parallel, the exploration of new feature generation is essential. This involves the creation of novel features and their comparison with the original set through automatic feature selection techniques. Mathematical operations for feature generation are tailored based on visual feedback and past experiences, enriching the feature space and potentially capturing latent patterns more effectively. Throughout this iterative process, continuous evaluation is conducted to assess the impact of feature engineering on model performance. Standard validation metrics such as accuracy, precision, and recall are leveraged for this purpose. Additionally, model performance is dynamically monitored, tracking the movement of instances based on updated prediction probabilities. This enables timely adjustments and enhancements to the modeling pipeline, ensuring the model's adaptability and efficacy in real-world scenarios.
In the filter approach, features are selected based on their intrinsic properties, independent of the model being used. Statistical measures like correlation, mutual information, or significance tests are employed to rank features according to their relevance to the target variable. Features meeting predefined criteria are retained, while others are discarded. This method is computationally efficient but may overlook interactions between features.
The wrapper approach evaluates feature subsets by training and testing models iteratively. It uses performance metrics, such as accuracy or cross-validation score, to select the optimal subset of features. This method considers interactions between features but can be computationally intensive, especially for large feature spaces. Despite its computational cost, wrapper methods tend to yield high-performance models as they directly assess feature subsets within the context of the chosen model.
Embedded methods integrate feature selection within the

model training process itself. Algorithms like LASSO (Least Absolute Shrinkage and Selection Operator) or decision trees inherently perform feature selection as part of their optimization process. These methods penalize or prune features based on their contribution to model complexity or predictive performance. Embedded methods strike a balance between filter and wrapper approaches, offering computational efficiency while considering feature interactions.

Hybrid approaches combine elements of filter, wrapper, and embedded methods to leverage their respective strengths. For instance, a hybrid method might initially use a filter approach to reduce the feature space based on statistical measures, followed by a wrapper approach to fine-tune feature subsets using iterative model evaluation. Alternatively, it could integrate embedded feature selection within a wrapper-based model training process. Hybrid methods aim to achieve the best of both worlds, balancing computational efficiency with model performance.

IV. Experimental Results

The most important step is dataset balancing,the dataset used in the project is "European Cardholders 2013 dataset" the first approach to data balancing was tabgan [4],however the method didn't provide the desired output, later the SDV library and the CTGAN Synthesiser was used to balance the dataset . feature selection is an important part of modelling , which involves experimenting with various techniques and strategies. Evaluation comes after feature selection and model training; after the model is trained, its efficiency is determined by calculating accuracy, precision, and f1 scores. This evaluation is common to all models used, which will be discussed in detail.Following the training of various models, a comparison is done between the various models trained and the methodology used in the feature engineering stage.

A. Dataset

The project utilised the European Cardholders 2013 dataset for the study, which is a reputable and useful source in the field of credit card fraud detection. With more than 280,000 transactions, this dataset is sizable and offers a big amount of data for building reliable machine learning models. The dataset's diversity in terms of transaction amounts and types is especially helpful because it helps the model learn and adapt more effectively to a range of real-world events. Furthermore, the dataset has already been transformed using Principal Component Analysis (PCA). By concentrating on the most pertinent features, this preprocessing stage helps to reduce noise and redundancy in the data, which may improve model performance.

B. Tabgan Mediated Approach

Table-to-Table Generative Adversarial Networks, or TabGANs, are an effective way to create synthetic data while maintaining the connections between the features in your original dataset.TabGAN is quite good at figuring out the general structure and connections in your data. It examines the relationships between several characteristics (quantity, place, and time) and records those dependencies in the data that is produced. However, it doesn't explicitly target specific class distributions.Since initially at first there were only fewer fraudulent transactions, the generated dataset also maintained the exact ratio and reflected the same imbalance as the original dataset.

Tabgan treated all data points as equals, this led to the model focusing on replicating data pattern within each class rather than adjusting the overall class proportions.This is where CTGAN entered the picture, allowing conditional sampling to be carried out and producing the necessary results by defining the various constraints and giving the various class objects varying degrees of priority.

C. Working of SDV and CTGAN Synthesizer

Firstly, the SDV library provides functionalities to create synthetic versions of structured datasets[6], which is particularly useful in scenarios where the original dataset contains limited instances of specific classes or where privacy concerns restrict direct access to real data both these characteristics matches our dataset . The process typically begins by defining the metadata for the dataset. Metadata includes information about the structure, characteristics, and dependencies within the dataset, such as column types, distributions, and relationships between features. Following the definition of the metadata, the SDV library uses a variety of generative modelling methods to discover the fundamental structure and statistical characteristics of the original dataset. In order to generate synthetic tabular data, we employ the Conditional Tabular Generative Adversarial Network (CTGAN)[5]. CTGAN uses adversarial training to train a generative model, which enables it to discover the intricate patterns and connections seen in the data. The model's goal during training is to produce artificial data samples that, when evaluated by a discriminator network, are identical to real data instances.

The requirements for creating synthetic data samples are then specified in the conditions for data sampling. For example, There's a need to designate conditions to sample a given number of rows when particular columns—like class labels or categorical values—meet predetermined standards. These constraints help ensure that the generated synthetic data closely reflects the actual dataset in terms of relevant features and distributions. In order to serve our aim, we need to construct more objects that have a "Class" value of "1,"[2] signalling the transaction is fraudulent.

The CTGANSynthesizer[6] object, which is in charge of training the CTGAN model and producing synthetic data, is initialised after the metadata and sampling circumstances are specified. To ensure that the generated data matches the original dataset's properties, the CTGANSynthesizer uses the sampling conditions and defined metadata to direct the production process. The CTGAN model is then trained after the CTGANSynthesizer has been fitted to the original dataset. Through iteratively adjusting its parameters, the CTGAN model generates synthetic data samples that closely mimic the original dataset while learning the underlying data distribution

and dependencies during training. After being trained, the CTGANSynthesizer can produce synthetic data samples for sharing data while protecting privacy, enhancing data, and testing models, among other uses.Adjustments to parameters and configurations can be made based on specific requirements and characteristics of the dataset.

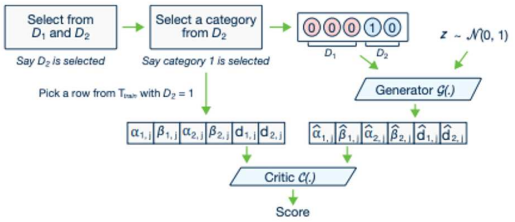

Fig. 3. CTGAN model. The conditional generator can generate synthetic rows conditioned on one of the discrete columns. With training-by-sampling, the cond and training data are sampled according to the log-frequency of each category, thus CTGAN can evenly explore all possible discrete values

Sampling with conditions and without conditions have their advantage and disadvantage.To create artificial data depending on predetermined criteria, utilise predefined function. By defining criteria for data sampling, these conditions enable the creation of artificial samples that satisfy particular needs. Here the conditions are specified ,target specific "fraud values" as a condition in this instance, the parameter for selecting fraudulent transactions are defined here . This guarantees that cases of fraudulent transactions are included in the synthetic data that is generated , enabling the addition of synthetic fraud occurrences to the dataset.

The dataset thus generated contains considerably a higher percentage of fraud transaction than the actual dataset.However further processing is absolutely required for the generated dataset before feature engineering experiments are carried on it .

Fig. 4. Graphical representation of how the data distribution varies in the real dataset and the synthetic dataset

D. Handling Imbalanced Datasets: Techniques for Fair Classification

As previously noted, there were more fraudulent transactions in the dataset created by the CTGAN Synthesizer than transactions that were legitimate.The issue here is that, as a result of this outnumbering, the machine can start to favour one of them and stray from recognising the fundamental patterns connected to both legitimate and fraudulent transactions.
The way CTGAN works is by examining the original data's statistical characteristics. In essence, it finds patterns and connections among many properties (such as transaction amount, location, and time) and tries to include those patterns into the artificial data it creates.CTGAN may give priority to reproducing the distribution of fraudulent transactions in the generated data if the original dataset has fewer of them than valid ones.This may result in synthetic data having more fraudulent transactions than authentic as is the case here.

To get around this problem, an effective solution was required. A number of post-processing approaches were investigated, and merging the created dataset with the original dataset resulted in a logical and effective way to get a suitable proportion of fraud cases to real ones[9].Combining the original and synthetic data sets provides a larger pool of data for training the model. In addition to allowing the model to learn from those patterns more successfully, this also contains more instances from the fraudulent class as well.By combining data from two sources, potentially introduce a wider variety of patterns and scenarios for the model to learn from[10]. This can lead to better generalizability, as the model is less likely to be overly reliant on the specific characteristics present in either the original or synthetic data alone.

E. Feature Selection

After the data balancing process is finished, we use a variety of feature selection techniques to extract useful information from the dataset. This may improve accuracy, reduce the dimensionality of the data, potentially mitigating over-fitting

1) Mutual Information : The measure of mutual dependency between two variables is called mutual information, or MI. When choosing features, the knowledge gained about the target class—in this case, fraudulent/legitimate transactions—is quantified. Higher MI scores are indicative of more informative features. In our research, we found the mutual information gain by using the sklearn library[11]. sklearn, To calculate mutual information for a supervised classification issue, utilise themutual_info_classif function found in the sklearn.feature_selection package. The mutual information for every pair of discrete variables (feature and class) is calculated by this function.

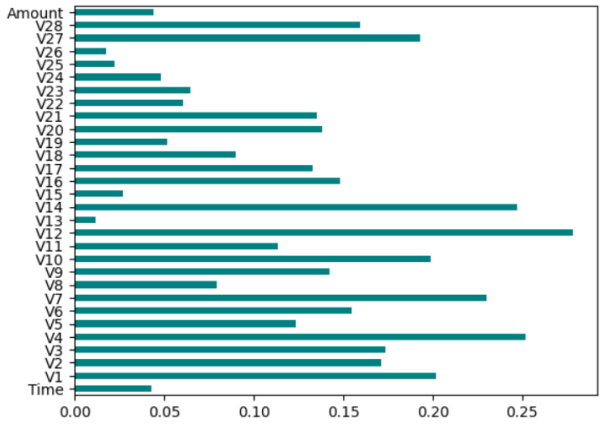

Fig. 5. Features and their information gain

2) Fisher Score: Through the calculation of the ratio between the inter-class and intra-class scatters, each feature's discriminative potential is assessed using the supervised feature selection approach . When attempting to identify the characteristics that most effectively distinguish the two groups, it is especially helpful in binary classification situations.

When implementing the Fisher score for feature selection, you typically calculate the score for each feature and then rank or select the top features based on their scores.The function fisher_score, provided by skfeature.function.similarity_based, can be used to determine the Fisher score for non-negative.

Both the methods where used and the inferences are based solely on the provide values and the context of a synthesized dataset.MSE,R-squared and Accuracy are calculated separately for the same dataset applying different feature selection method MSE and R-squared are better suited for regression tasks,these metrics measure how well a model predicts continuous values.Fraud detection is a classification task,here we're are classifying transaction as fraudulent or legitimate

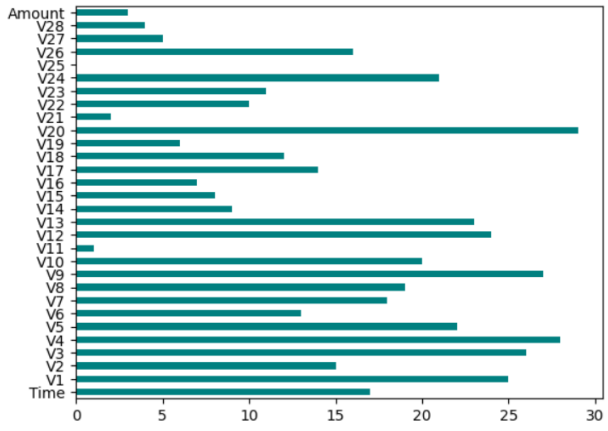

Fig. 6. Features and their fisher score

On detecting credit card fraud although Fisher Score shows a Mean Squared Error (MSE) and a higher R squared these metrics are more appropriate for regression tasks. In the case of fraud detection, which involves classifying fraudulent versus legitimate transactions accuracy is crucial. This is where Information Gain excels. It delivers accuracy indicating a number of correctly classified transactions. Therefore Information Gain is a choice for selecting features in the project as it prioritizes features that play a role, in accurately distinguishing fraudulent transactions from legitimate ones.

V. Implementation

A. CTGAN Synthesizer and Conditional Sampling

If we focus much on creating data it could lead to an imbalance, in the synthetic data. To prevent this CTGAN provides the option to set criteria, for the generated data. It's important to maintain a mix of both valid transaction scenarios to ensure an outcome. Keep in mind that incorporating scenarios is essential to avoid generating synthetic data. Of the many strong reasons, a key aspect of CTGAN is its conditional sampling capability.This enable the user to specify conditions into the generated data.

Functioning of Conditional Sampling

Step 1: 1) Defining Conditions: We can define specific conditions for the generated data. These conditions can target various attributes like transaction amount, location, time of day, etc.

Step 2: 2) Targeting Legitimate Transactions: Importantly, we can specify a condition where the class label has a value of "1," indicating a fraudulent transaction. By including this condition, CTGAN will only generate synthetic data points that fall under the category of legitimate transactions.

Step 3: 3) Sampling with Conditions: During the data generation process, CTGAN will utilize these defined conditions to ensure the newly created data points adhere to the specified criteria. This way, we can generate realistic and legitimate synthetic transactions.

B. Data Merging and Data Balancing

The CTGAN Synthesized data had more fraud samples and in order to balance the dataset and to ensure a balanced representation of both type that is real-world dataset that likely contains more legitimate transaction and the generated dataset which has be been generated by determining the pattern associated with the fraud ones.

The model(Trained using original dataset + synthetic data) achieved a score of 98 for accuracy, suggest the model learned effectively to distinguish between fraud and legitimate transaction with high accuracy.When the same model was tested against a new dataset generated using CTGAN the model showed a surprising accuracy of 92 and satisfiable r2 score and mse values as well.

Even with advanced techniques like Generative Adversarial Networks (GANs), feature engineering remains a cornerstone of building high-performing fraud detection models

C. Mutual Information Feature Ranking and Modelling

The significance of Different Features where evaluated using the Information gain ,Mutual information between two random variables is a non negative value,it is equal to zero if and only if two random variables are independent,and higher values mean higher dependency.
The function relies on non parametric methods based on entropy estimation from k-nearest neighbors distances as described in [12] and [13]. Both methods are based on the idea originally proposed in[14].
Based on the mutual information gain a rank was made between the features and the best features where selected and the the dataset was constricted this reduces the overall size and complexity of the original dataset keeping the overall essential characteristics and relationship within the original data while eliminating redundant or less informative features and the the model was trained on this reduced dataset.
The model turned out to be excellent with a mean squared error of 0.79 and an accuracy of about 95 percentage . By prioritizing features with high Information Gain, we ensure the model focuses on the most significant aspects of the data. This not only reduces training time but also improves the model's predictive performance by providing it with the most relevant information to distinguish between legitimate and fraudulent transactions.

VI. CONCLUSION

In this study, we addressed the challenge of credit card fraud detection using a highly imbalanced dataset from European cardholders in 2013. The significant imbalance, with only a small number of fraudulent transactions, could have hindered the model's ability to effectively learn and identify fraudulent patterns.
To mitigate this issue, we implemented a Conditional Generative Adversarial Network (CTGAN) approach. CTGAN's conditional sampling capability allowed for the generation of synthetic legitimate transactions that resembled the real data. By merging this synthetic data with the original dataset containing real fraudulent transactions, a more balanced dataset was created for model training.
Furthermore, we employed information gain for feature selection. This technique ranked features based on their relevance to predicting fraudulent transactions. The top-ranked features were then used to create a reduced dataset, maintaining the essential characteristics of the original data while improving training efficiency. Various machine learning models were subsequently trained on this reduced dataset.
The initial results were promising, with the model achieving high accuracy on the combined dataset, indicating its ability to distinguish between legitimate and fraudulent transactions. However, a drop in efficiency was observed when tested on new CTGAN-generated data. This highlights the importance of addressing potential distribution shifts and exploring techniques such as model fine-tuning or ensembling for improved generalizability on unseen data.
Summing up, this study demonstrates the effectiveness of combining data balancing techniques like CTGAN with feature selection methods like information gain to build robust credit card fraud detection models. By acknowledging limitations and refining approaches, we can strive to create models that are not only accurate but also generalize well to real-world scenarios, ultimately leading to a more secure financial environment.

VII. STATEMENTS AND DECLARATIONS

Competing Interests: On behalf of all authors, the corresponding author states that there is no conflict of interest.

Funding Information: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Author Contribution: The work was conceptualized by Anand D, Afeef Ashraf C, and Ajaz Ali; Anand D created the methodology; Afeef Ashraf C and Ajaz Ali carried out the experiments; Anand D, Afeef Ashraf C, and Ajas Ali examined the data and wrote the manuscript; Shabiya M I supervised the research. All of the direction and correction assistance came from Rini T. Paul.

Data Availability: The data that support the findings of this study are openly available at https://drive.google.com/drive/folders/1-COAGylmkaoWND_HFS_BwbOpozqRhaZL?usp=drive_link and https://github.com/luci123fer456/random.git

Research Involving Human and/or Animals: This research did not involve human participants or animals.

Informed Consent: Not Applicable

REFERENCES

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, *Generative adversarial networks*, 2014. arXiv: 1406.2661 [stat.ML].

[2] F. A. Almarshad, G. A. Gashgari, and A. I. A. Alzahrani, "Generative adversarial networks-based novel approach for fraud detection for the european cardholders 2013 dataset," *IEEE Access*, vol. 11, pp. 107 348–107 368, 2023. DOI: 10.1109/ACCESS.2023.3320072.

[3] T. Rawat and V. Khemchandani, "Feature engineering (fe) tools and techniques for better classification performance," May 2019. DOI: 10.21172/ijiet.82.024.

[4] I. Ashrapov, "Tabular GANs for uneven distribution," *Preprints*, Oct. 2020. arXiv: 2010.00638 [cs.LG].

[5] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," 2019.

[6] N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault," in *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct. 2016, pp. 399–410. DOI: 10.1109/DSAA.2016.49.

[7] A. Chatzimparmpas, R. M. Martins, K. Kucher, and A. Kerren, "Featureenvi: Visual analytics for feature engineering using stepwise selection and semi-automatic extraction approaches," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 4, pp. 1773–1791, 2022. DOI: 10.1109/TVCG.2022.3141040.

[8] A. Nugroho, A. Z. Fanani, and G. F. Shidik, "Evaluation of feature selection using wrapper for numeric dataset with random forest algorithm," in *2021 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2021, pp. 179–183. DOI: 10.1109/iSemantic52711.2021.9573249.

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Oversampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, DOI: 10.1613/jair.953. [Online]. Available: https://doi.org/10.1613/jair.953.

[10] F. Rodríguez-Torres, J. F. Martínez-Trinidad, and J. A. Carrasco-Ochoa, "An oversampling method for class imbalance problems on large datasets," *Applied Sciences*, vol. 12, no. 7, 2022, ISSN: 2076-3417. DOI: 10.3390/app12073424. [Online]. Available: https://www.mdpi.com/2076-3417/12/7/3424.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[12] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E*, vol. 69, p. 066 138, 6 Jun. 2004. DOI: 10.1103/PhysRevE.69.066138. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.69.066138.

[13] B. C. Ross, "Mutual information between discrete and continuous data sets," *PLoS ONE*, vol. 9, no. 2, D. Marinazzo, Ed., e87357, Feb. 2014, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0087357. [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0087357.

[14] L. H. Schick, "Two recent applications of maximum entropy," in *Maximum-Entropy and Bayesian Spectral Analysis and Estimation Problems*. Springer Netherlands, 1987, pp. 283–293, ISBN: 9789400939615. DOI: 10.1007/978-94-009-3961-5_17. [Online]. Available: http://dx.doi.org/10.1007/978-94-009-3961-5_17.