National College of Ireland

Predicting Credit Card Fraud Using Conditional Generative Adversarial Network

MSc Research Project
Data Analytics

Purnima Duggal
Student ID: x20237928

School of Computing
National College of Ireland

Supervisor: Vladimir Milosavljevic

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Purnima Duggal
Student ID:	x20237928
Programme:	Data Analytics **Year:** 2022
Module:	MSc. Research Project
Supervisor:	Vladimir Milosavljevic
Submission Due Date:	19-09-2022
Project Title:	Predicting Credit Card Fraud Using Conditional Generative Adversarial Network

Word Count 6395 **Page Count** 19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: .....

Date: .....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Predicting Credit Card Fraud Using Conditional Generative Adversarial Network

Purnima Duggal
x20237928

Abstract

As the financial sector continues to flourish over these years, there have been significant changes in many conventional systems. One of them is carrying money in hand. Credit cards and debit cards have completely replaced carrying money in hand as the predominant method of payment. Numerous studies indicate that the use of credit cards vastly increased after the pandemic period. The increasing use of credit cards is a common target for cybercriminals and fraudsters. Although numerous steps have been taken to prevent credit card fraud, the problem still exists. The skewness or imbalance of the dataset is the most frequent problem that researchers encounter when conducting analysis using various machine learning models. In this research, I have addressed the problem of skewness of the dataset by using two approaches: SMOTE Technique and an unsupervised machine learning technique that is CT-GAN (Conditional Generative Adversarial Network). We used AUPRC as a performance indicator for both approaches. Three classifier models namely: Isolation Forest, Multi-Layer Perceptron and Random Forest are used to perform the experiments. We discovered that GAN performs well on two out of three models after analyzing both approaches and Isolation Forest outperforms the other two models, correctly detecting 86 % of credit card fraud.

1 Introduction

1.1 Background

Card-based payment systems have proven to be a better alternative than cash payments, as they handle many risks such as petty crimes associated with carrying cash in your pocket. Credit card companies offer multiple economic advantages to their consumers such as cost spreading in periodic payments, loyalty points for every purchase, travel points, no interest charges, no transaction fees, and many more. Due to these advantages, the credit card demand has increased significantly, especially during the covid pandemic era people tend to spend more money through credit cards as compared to cash, as stated by a recent article[1], also the people fear the risk of touching the surfaces while purchasing something from cash and credit card online payments proved to be the safest option when making any big purchases. However, even after the covid times, people continue to spend through their cards as stated in a recent economic study done on Irish citizens, the study shows that there is a 3 percent rise in credit card transactions after the covid era. As everything has its pros and cons, the credit card sector also comes with scams and frauds. Credit card fraud is increasing day by day and has become one of the biggest challenges to handle for many banks, financial organizations, and governments.

[1] https://www.ualberta.ca/folio/2020/08/why-not-using-cash-during-covid-19-could-make-you-more-likely-to-overspend.html

These fraudsters often target people who are engaged in some online transaction and empty all their bank accounts within some hours.
Credit card fraud takes place when an unauthorized individual accesses the credit card without approval from the cardholder. These card-related thefts are sometimes not noticeable for many days to some people which gives a major advantage to these fraudsters and makes the chance of catching the perpetrators more difficult.

1.2 Global Statistics on Credit Card Fraud

Many nations are struggling with the challenge of credit card fraud. It has become a huge concern for many nations. Till the year 2020, 459,297 credit card fraud has been reported. As with the growing technology, nowadays fraudsters are using sophisticated scams, they extract detailed personal information from cardholders and then employ that data to take control of already-existing accounts or to make brand-new accounts for fictitious individuals. Most fraudsters also use phishing emails to get into people's bank accounts by extracting confidential information.
In a recent report[2], over the last few years, the United States has been the largest recipient of credit card fraud from all over the world due to its more prevalent use of credit cards which accounts for more than one-third of all the losses globally. Ireland, France, and the UK are the top three European nations for credit card fraud after the US. The Nilson report states that bank losses exceeded $28.58 billion globally in 2020. In the next five years, global fraud will rise up to $43 billion.
Hence, it is crucial to develop fraud detection techniques to prevent such huge losses to banks and other financial institutions. Despite the development of various other techniques by banks such as a card verification system for each transaction, new security measures adaptation in addition to a new payee, cybercriminals continue to outsmart security systems by evolving over time.

1.3 Research Question

A credit card detection technique can be challenging because of the class imbalance of credit card datasets. Many machine learning techniques are used by various researchers for rectifying this issue, still, the issue of an imbalanced dataset continues to be a hindrance, they get a high degree of accuracy in results however due to low numbers, the minority class is neglected in that scenario. It is crucial to take advantage of research studies that have taken into account this issue. Despite the existence of numerous traditional methodologies, the class imbalance nature of the dataset is given little consideration.
GAN is applied in different additional machine learning areas and has given significant outcomes in various research studies. To address the issue of imbalance, I have proposed a novel technique that combines Conditional GAN with other machine learning models including Isolation Forest, Multilayer Perceptron, and Random Forest.
Therefore, the research question for this project is:

"How can Conditional General Adversarial Network (GAN) be used in combination with other machine learning models to more accurately predict credit card fraud and solve the issue of imbalance dataset?"

This research paper is structured into these sections:

[2] https://chargebacks911.com/credit-card-fraud-statistics/

- Section 2(Related Work): This section delves into previous credit card fraud detection methods and offers an in-depth analysis of strategies and methodologies used in various studies to forecast credit card theft by taking into account a range of factors.

- Section 3(Methodology): This section describes the approach utilized in this study to generate the results. This section also includes a few subsections that break down the process step-by-step.

- Section 4(Design Specification): This section depicts the architectural design of this research. It contains a description of the approach followed.

- Section 5(Implementation): This section describes the implementation procedure thoroughly. It has several sub-sections that depict the steps followed to get the required results.

- Section 6(Evaluation): This section explains the numerous experiments conducted for this study and the evaluation standards that were applied.

- Section 7(Conclusion): This section outlines the final statements and the possibilities for future work in the domain.

2 Related Work

Machine learning is commonly regarded as an efficient and promising method for identifying or forecasting fraud. Almost all the datasets in the research studies were plagued by two serious issues. In the first place, there is a substantial class imbalance, as the number of fraud transactions is much less than that of non-fraud transactions. The second issue is the sequential nature of datasets. Analyzing and critically evaluating the currently available literature in this domain is vital for moving forward with the methodology. Our evaluation of the prior works can give us insight into their limitations or shortcomings, issues they faced while doing analysis, as well as how they could be improved in our research.

2.1 Existing Techniques and Methodologies

Several studies have been carried out in this area. Benchaji, I., and Douzi (2021) conducted research utilizing various machine learning techniques to forecast credit card fraud, namely LSTM networks that are used for long-term dependencies along with action patterns in order to organize items that are most significant to a classification problem, SMOTE technique to solve imbalanced data, UMAP to reduce dataset size. The author also uses SVMs and ANNs for comparative analysis and uses performance metrics to evaluate his results.

Khatri(2020) conducted similar comparison research using machine learning algorithms that include Decision trees, KNN, Logistic regression, and the Naive Bayes algorithm. The authors used a public dataset and filtered 28 variables from it. It was concluded that the decision tree had a higher accuracy rate (85.1%) than any other model.
Dr. K. Kaur(2021) overcame Khatri's (2020) drawback by employing a resampling method called the Near Miss method, which directs the division of the data into separate segments.

The author performs an analysis of the decision tree and Naive Bayes technique, with the help of the confusion matrix, demonstrating that the near miss method is able to balance the dataset and prevent data loss. With an accuracy of 97 percent, decision trees outrank Naive Bayes. The author concluded that neural networks can be used to improve the model's performance.

The study by Negi S. (2022) focuses on how two supervised techniques—Multi-Layer Perceptron, Logistic and XGBoost—can handle the issue of large datasets. The author conducted various experiments and concluded that Multi-Layer Perceptron outperforms other models.

Zadafiya et al (2022) research depend on two algorithms: isolation forest and local outlier factor. Although the isolation forest performs well in the classification report, the author finds that the class imbalance problem remains unresolved.

Forough J., Momtazi(2021) employs a probabilistic model approach, taking into account the sequential nature of the data by combining the LSTM(Long short-term memory) and CRF(Conditional Random Field) model. After processing, the LSTM obtains a series of transaction data as input and provides output arrays; this output is then assigned to the CRF layer as its input. As a result, a decision is made based on the probability of successful predictions by the CRF layer, and the author uses under-sampling techniques to account for data sequences.

2.2 Examining the Class Imbalance Issue

As mentioned above, one of the primary issues that researchers encountered in this research is a highly skewed dataset. A.Guzman-Ponce(2021) offers a dual-level solution to tackle with the issue of class imbalance. He first implements the DBSCAN clustering algorithm that extracts approximately half of the dataset from noisy data, and a graph-based method to handle the highly skewed nature of data. He then carried out his research on 24 different datasets and employs geometric mean as performance metrics. The author employs various techniques, such as the Clustering approach and random under-sampling, and then compares the performance of these techniques to draw conclusions. The author concludes that DBSCAN performs well on 19 out of 24 datasets.

S.Makki(2019) also employs two approaches to deal with data imbalance; first, he adopts the oversampling and under-sampling techniques, and then he follows the Cost-Sensitive Approach. The author conducted numerous experiments using eight different machine learning techniques. It was concluded by the author based on three performance measures, Accuracy, AUPRC, Sensitivity, that SVM, and ANN are more effective in predicting credit card fraud.

Brennan, P. (2012) also incorporates two different approaches, such as under-sampling, oversampling, and SMOTE, into his approach to ensure data balance between classifiers; he uses the RIPPER algorithm and the RoS approach to accomplish this; SMOTE proves to be the best, while RIPPER improved the results.

2.3 Generative Adversarial Network

Among the many applications of GAN, A. Aggarwal et al. (2021) discuss how GAN is used in domains such as image analysis, facial detection, traffic management, and also describes different models that use GAN. The authors also describe how A 3D object generation algorithm uses GAN as an adversarial architecture that produces high-quality images, and how visuals of traffic helps can be used by the GAN model to manage the traffic. The GAN structure and problems encountered by important domains are also covered by the author.

In their study on GAN, Ngwenduna and Mbuvha(2021) discussed how GAN is more efficient at coping with class imbalances and is more flexible than other approaches. Furthermore, the authors compared GAN to other existing techniques, such as SMOTE, and discovered that while GAN was more reliable for some datasets, it had drawbacks for others, which SMOTE handled well. The author also talks about numerous other GAN kinds and under what circumstances they should be used, for instance, WGAN overcomes GAN model restrictions.

This research paper by H. Liu et al.(2022) proposes the use of GAN in combination with bi-LSTMs and AE in modeling machine health monitoring systems. According to the author's proposed model, there are three stages, the first is the forecast of reconstruction and the training of distinct representations by GAN, followed by filtering of key features and dimension reduction by AE. The final step is to build the supervised learning model and integrate it with feature data to forecast the diagnosis.
As opposed to the other research investigations, Gui, J., (2020) explains the GAN approach from a mathematical standpoint. The author offers three different approaches to train GAN model. The GAN's structure is covered in the first step, followed by the combination that must be integrated to the GAN and the third is the kind of application that will be using it. The author also discusses several GAN model constraints and how to get around them in various scenarios. He also discusses other GAN model variations.

3 Methodology

Data mining employs a wide range of methodologies, which includes CRISP-DM, KDD, and SEMMA. We have used CRISP-DM (Cross Industry Process for Data Mining) methodology in this research. The below diagram depicts the six steps of the CRISP-DM methodology.

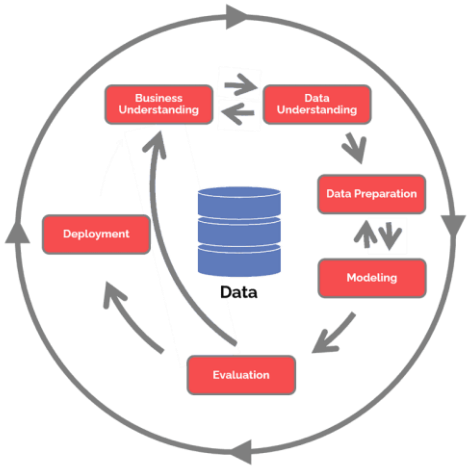


Figure 1: Six Steps of CRISP-DM, Image taken from Source[3]

3.1 Business Understanding

Business Understanding is the first stage of the CRISP-DM approach. The main aim of this project is to forecast fraudulent and legitimate transactions. Our objective is to create a model that achieves this main objective while taking the imbalanced dataset's constraints into account. The Conditional GAN model is being used to address the imbalance dataset problem

[3] https://www.datascience-pm.com/crisp-dm-2/

since it can produce synthetic data that the model can train on before being tested against actual data. We can address the global issue of credit card theft with this unique combination of Conditional GAN with machine learning models including Isolation Forest, Multilayer Perceptron, Ada Boosting, and Random Forest. Financial institutions and banks working with credit card fraud issues would greatly benefit from this research since it will help them cut down on credit card scams.

3.2 Data Understanding

Data Understanding is the second stage in CRISP-DM. Before building the model, it is necessary to have a thorough understanding of the dataset that includes all elements, features, data types, their correlation with each other, percentage of minority vs majority, and many more. This hugely assists us in the creation of an efficient model.

We have considered data[4] from an open source Kaggle repository, where the data is publicly available and the information is masked so no personal information of any individual is affected.

The dataset contains credit card transactions done in two days in September 2013. The dataset contains 30 features, 28 of which are coded from V1 to V28 in form of PCA components for confidentiality except for Time and Amount. Only one feature is categorical; the rest are continuous in nature. The Time variable in the dataset represents the transaction time between each transaction and the Amount variable represents the credit or debit Transaction Amount.

3.2.1 Data Exploration

Data exploration is useful for understanding the nature of data elements, as well as for identifying majority and minority elements in data through various visualizations.

- **Determining the Data Type of each variable**

  We will begin by determining the data type of our dataset's variables. As seen in figure 2 below, all the variables are float type, only the target variable is an integer.

  We do not need to perform any conversion for our variables and we can move to the next step.

```
Sample size : 284807
<class 'pandas.core.frame.DataFrame'>
Int64Index: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64
dtypes: float64(30), int64(1)
memory usage: 69.5 MB
```

[4] https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Figure 2: Data Types of each variable

- **Examining missing and redundant data**

As a next step, we will check the dataset for missing values in both rows and columns.

A. Missing values in the dataset

On analysing we found that there are no missing values in the rows or column in the dataset.

(a) Missing values in columns (b) Missing values in rows

Figure 3: Missing values

B. Duplicate values

```
# Check for duplicates

duplicate = df[df.duplicated()]
duplicate
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
33	26.0	-0.529912	0.873892	1.347247	0.145457	0.414209	0.100223	0.711206	0.176066	-0.286717	...	0.046949	0.208105	-0.185548	0.001031	0.098816	-0.552904	-0.073288	0.023307	6.14	0
35	26.0	-0.535388	0.865268	1.351076	0.147575	0.433680	0.086983	0.693039	0.179742	-0.285642	...	0.049526	0.206537	-0.187108	0.000753	0.098117	-0.553471	-0.078306	0.025427	1.77	0
113	74.0	1.038370	0.127486	0.184456	1.109950	0.441699	0.945283	-0.036715	0.350995	0.118950	...	0.102520	0.605089	0.023092	-0.626463	0.479120	-0.166937	0.081247	0.001192	1.18	0
114	74.0	1.038370	0.127486	0.184456	1.109950	0.441699	0.945283	-0.036715	0.350995	0.118950	...	0.102520	0.605089	0.023092	-0.626463	0.479120	-0.166937	0.081247	0.001192	1.18	0
115	74.0	1.038370	0.127486	0.184456	1.109950	0.441699	0.945283	-0.036715	0.350995	0.118950	...	0.102520	0.605089	0.023092	-0.626463	0.479120	-0.166937	0.081247	0.001192	1.18	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
282987	171288.0	1.912550	-0.455240	-1.750654	0.454324	2.089130	4.160019	-0.881302	1.081750	1.022928	...	-0.524067	-1.337510	0.473943	0.616683	-0.283548	-1.084843	0.073133	-0.036020	11.99	0
283483	171627.0	-1.464380	1.368119	0.815992	-0.601282	-0.689115	-0.487154	-0.303778	0.884953	0.054065	...	0.287217	0.947825	-0.218773	0.082926	0.044127	0.639270	0.213565	0.119251	6.82	0
283485	171627.0	-1.457978	1.378203	0.811515	-0.603760	-0.711883	-0.471672	-0.282535	0.880654	0.052808	...	0.284205	0.949659	-0.216949	0.083250	0.044944	0.639933	0.219432	0.116772	11.93	0
284191	172233.0	-2.667936	3.160505	-3.355984	1.007845	-0.377397	-0.109730	-0.667233	2.309700	-1.639306	...	0.391483	0.266536	-0.079853	-0.096395	0.086719	-0.451128	-1.183743	-0.222200	55.66	0
284193	172233.0	-2.691642	3.123168	-3.339407	1.017018	-0.293095	-0.167054	-0.745886	2.325616	-1.634651	...	0.402639	0.259746	-0.086606	-0.097597	0.083693	-0.453584	-1.205466	-0.213020	36.74	0

Figure 4: Check for duplicate values

There are some duplicate values present in the dataset and those needs to be handled in data preparation phase as seen in figure 4.

- **Examining the Distribution of class**

Figure 5: Pie chart illustrating the distribution of fraud vs non-fraud transactions

There are about 284,315 total transactions out of which only 492 are fraud transactions as seen in Figure 5. Based on percentage, 0.167% of transactions are fraudulent, while 99.833% of transactions are non-fraudulent.

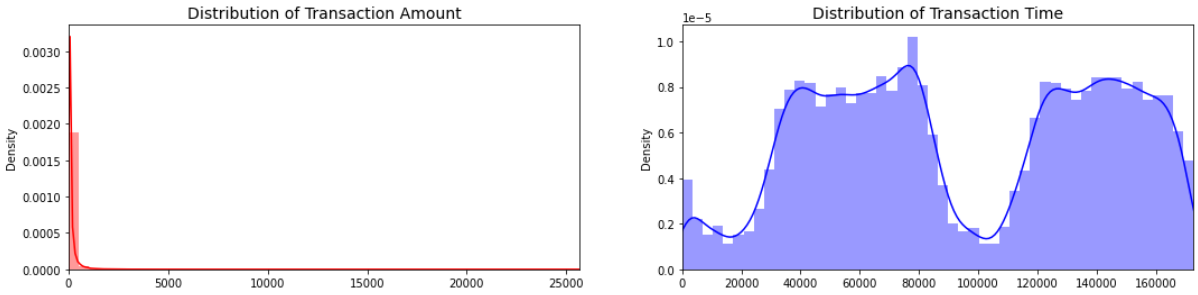


Figure 6: Distribution of Transaction Amount and Transaction Time

Based on Figure 6, we can observe that transaction amount and time distributions are highly skewed and should be handled during the data preparation process.

3.3 Data Preparation

This is also known as the Data Pre-Processing stage; it is the process of transforming raw information into data that can be immediately used to apply models. Since this stage comes before using any models, this is the most crucial step and helps in building scalable, robust, and efficient models. This phase entails dealing with missing or duplicate values, handling outliers by scaling variables, formatting data types, and feature engineering. Low-quality data preparation techniques may result in lengthy computations, high expenses, and poor model performance. Therefore, it is crucial to give equal importance to all of the factors involved in the data preparation step.

- **Dealing with Missing and Duplicate Values**: As illustrated in Figure 4, there is some duplicate data that must be managed. We removed the duplicate values from the dataset, and there were no missing values in rows or columns as seen in Figure 3.

- **Scaling of variables in the dataset:** We discovered the range scale variance between all other features other than Time and Amount by analyzing our data. Scaling needs to be done on Time and Amount, we have used the “RobustScalar” method that removes the outliers.

- **Identifying and addressing class imbalances:** It has been described in the above sections that class imbalance is one of the critical issue in credit card datasets. It can be noted from Figure 5 that our dataset is highly imbalanced having very less fraud transactions as compared to non- fraud transactions. So, before applying models, we need to take into account this issue and create such a dataset using GAN so that the no. of fraud transactions becomes equal and it is easy to apply other models. Several other algorithms are available to manage class imbalance of data, including under-sampling, subsampling, and SMOTE, but we chose GAN after reviewing several research papers and taking into account the advantages and disadvantages of other techniques.
  In this study, we present a novel strategy to deal with this issue of class imbalance by employing Generative Adversarial Networks (GAN). As discussed in the above sections, our dataset only consists of 492 fraud transactions out of a total of 284,315 transactions, so we need to produce synthetic data to train our models, with the use of GAN we will be generating this synthetic data.
  We would use this synthetic data to train our models and then once the models are trained, we will test our analysis on real data. GAN has several advantages over other models.

❖ Brief Overview of GAN

Generative Adversarial Networks are comprised of two neural deep learning networks that compete against each other. The main elements of a generative adversarial network include; A Generator and A discriminator. The generator G attempts to learn the data distribution by using random noise as input and producing realistic datasets as its output. Conversely, the discriminator D classifies whether the produced sample is from the genuine dataset or from a fake dataset, which is generated by the generator. Both the network work against each other, in the starting the generator produces gibberish data as it has no idea in the starting but as soon as the discriminator tries to give feedback to the generator not only does the generator tries to improvise the data but is able to produce more realistic data of the original dataset. GANs are said to be extremely effective at creating realistic new data samples that closely resemble our training-data distribution, and are proving to be a show stopper in the field of Machine Learning. The below figure illustrates the entire process of GAN architecture.

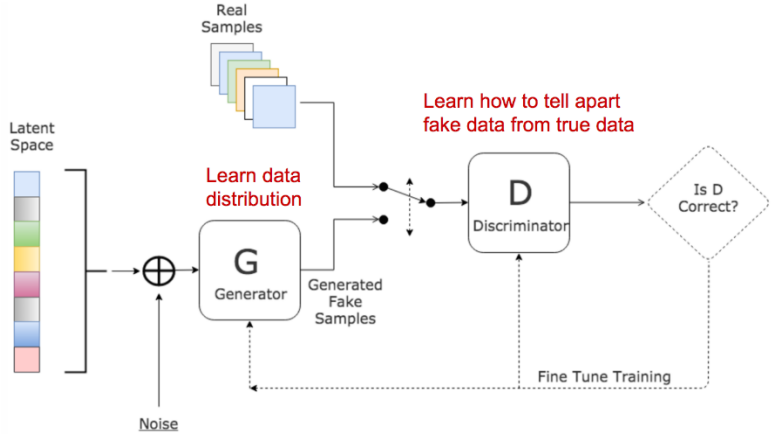


Figure 7: GAN Architecture, Image taken from source [5]

[5] https://zhongpeixiang.github.io/generative-adversarial-network-overview/

Issues related to GAN

The following significant issues in the generator and discriminator plague many GAN models.

- Non-convergence issue: GAN models suffer from this problem, in which the parameters fluctuate, become unsteady, and never converge, which compromises their performance and sometimes leads to underfitting of the model.

- Mode collapse: This typically happens when the generator produces inaccurate data based on the input noise, resulting in less variability in data and failing to account for the uneven nature of Data, therefore producing skewed data.

- Diminishing gradient: The discriminator becomes so successful that the generator is unable to learn any new information which leads to an inadequate augmentation of data.

Conditional GAN can solve all these issues related to GAN. The following subsection describes how CT-GAN operates.

❖ Conditional GAN (CT-GAN)

In CT-GAN[6], a conditional configuration is used both in generator and discriminator on class labels. It is a sophisticated framework built by taking the GAN architecture into consideration. As a result, by providing the contextual data, the model is able to learn more than one mapping from input to output. CT-GAN handles the challenge of losing information and overfitting.

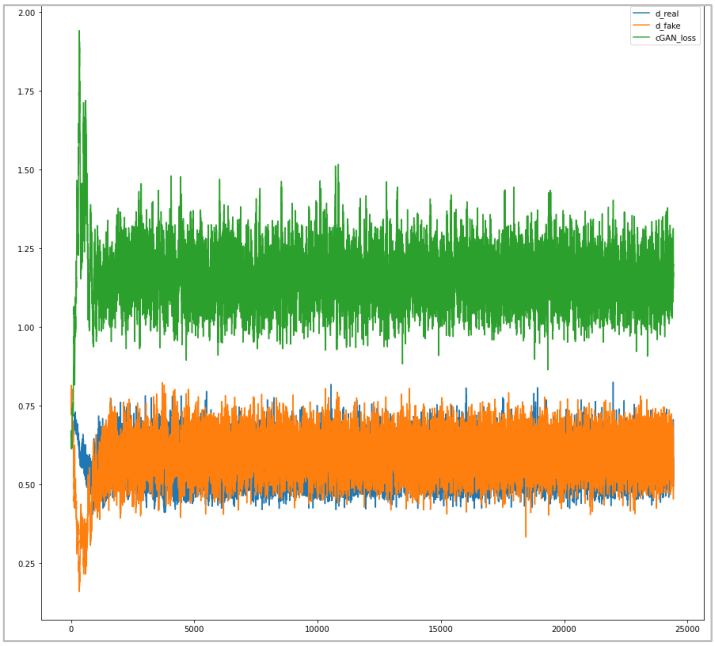


Figure 8: Data generated by CT-GAN

CT-GAN comprises of a conditional generator that helps to produce uniformly distributed data points in the training stage so that data distribution is maintained in the augmented instances.

[6] https://www.educative.io/answers/what-is-a-conditional-gan-cgan

Both the conditional generator and discriminator utilize the LSTM layers which allow them to record and retrieve all potential dependencies between the provided data. CT-GAN employs many parameters that affect model learning, these parameters directly affect the model performance, and quality of data points generated. The CT-GAN trains the model with the WGAN loss function which addresses the limitation of diminishing gradient of GAN. Also, it uses the RELU activation function and batch normalization.

3.4 Modelling

This is the most essential phase of CRISP-DM because it is where we apply various models to pre-processed data. Since the data is already in decent shape, we can apply different models to see which one best fits our dataset. So, this phase is broken down into four components: It starts with selecting the model, then generating a test design that is by creating a procedure for evaluating the accuracy and reliability of the model, then building the model, and finally assessing the model by various performance metric tools.
The following section discusses the various models used in this research.

- **Isolation Forest**
  Isolation Forest is a type of unsupervised machine learning model[7]. The model chooses or isolates an attribute at random and divides it into maximum and minimum values. This process is done iteratively in order to generate a decision tree. The no. of splits required to separate a sample is equal to the distance from the root node to the last node. We have created a dictionary models_dict[] to store all the models and call them after applying the CT-GAN to generate synthetic data.
  We created an Isolation Forest function called IsF() in which we defined the model, generated a test design, and converted the isolation forest predictions to our classification problem of detecting fraud and non-fraud transactions.

```
# Let's define the IsolationForest Model
def IsF (X, y, ratio) :

    # train test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 1)

    # define the model
    model = IsolationForest(n_estimators = 50, contamination = ratio, random_state = 1)

    # train the model
    model.fit(X_train, y_train)

    # predict on the test set
    predictions = model.predict(X_test)

    # Convert the predictions according to problem profile
    predictions[predictions ==  1] = 0
    predictions[predictions == -1] = 1

    # evaluate the accuracy and classification report
    print('IsF : ' + str(accuracy_score(y_test, predictions)))
    print(classification_report(y_test, predictions))

    # assign this to models_dict
    models_dict['IsF'] = model
```

Figure 9: Isolation Forest

- **Multi-Layer Perceptron**

[7] https://ieeexplore.ieee.org/document/8756130

Multi-Layer Perceptron is a supervised machine learning algorithm with at least three layers: input, hidden, and output. It is a feedforward neural network in which each node employs an activation function. We have defined a function MLP() for Multi-Layer Perceptron in which we have defined the model, trained the model and defined the evaluation matrix in this function. The no. of hidden layers value is passed once the data is generated by CT-GAN. This model is also added to the newly created Models Dictionary.

```
# Let's define the MultiLayer Perceptron
def MLP (X, y, param) :

    # train test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 1)

    # hyper-parameters
    nodes = param['nodes']
    lrate = param['lrate']
    toler = param['toler']
    batch = param['batch_size']

    # define the model
    model = MLPClassifier(hidden_layer_sizes = (nodes,), tol = toler, batch_size = batch, learning_rate_init = lrate,
                          verbose = 0, random_state = 1)

    # train the model
    model.fit(X_train, y_train)

    # predict on the test set
    predictions = model.predict(X_test)

    # evaluate the accuracy and classification report
    print('MLP : ' + str(accuracy_score(y_test, predictions)))
    print(classification_report(y_test, predictions))

    # assign this to models_dict
    models_dict['MLP'] = model
```

Figure 10: Multi-Layer Perceptron

- **Random Forest**

Random Forest is a supervised machine learning technique that addresses the overfitting issue by finding the final result on average or majority rating. In our function RFR(), we defined the model's definition, training, and assessment criteria. Gini impurity criterion is also added to the model to determine the probability of likelihood of categorizing incorrectly if chosen in an arbitrary manner.

```
# Let's define the Random Forest Classifier Model
def RFR (X, y, split) :

    # train test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 1)

    # define the model
    model = RandomForestClassifier(n_estimators = 100, criterion = 'gini', min_samples_split = split, random_state = 1)

    # train the model
    model.fit(X_train, y_train)

    # predict on the test set
    predictions = model.predict(X_test)

    # evaluate the accuracy and classification report
    print('RFR : ' + str(accuracy_score(y_test, predictions)))
    print(classification_report(y_test, predictions))

    # assign this to models_dict
    models_dict['RFR'] = model
```

Figure 11: Random Forest

4 Design Specification

Figure 12 shows the architectural design of our research. We are using data from the Kaggle repository. The selected data is first subjected to data processing, which includes the removal

of any duplicate values, or null values, and the scaling of the data using the RobustScalar method. As the dataset is highly imbalanced so CT-GAN is applied to the dataset so that more data points can be produced for the minority class of data.

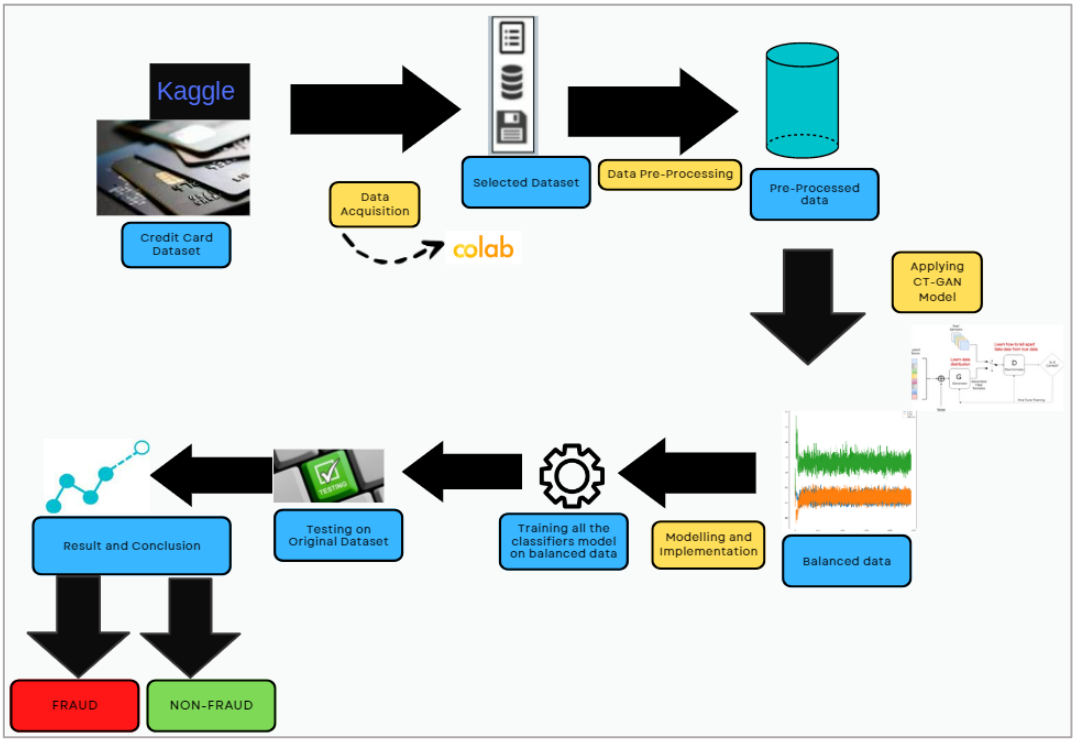


Figure 12: Architectural design of Credit Card Fraud using CT-GAN

.
When the synthetic data is ready, we combined it with the original dataset to create a new data set. In order to apply the models, we divide our data into training and testing after this phase. Now, we'll use each of the three models individually, applying the necessary parameters to the training set of data, and monitoring the outcomes. We have created separate functions for all the models: Isolation Forest, Multi-layer Perceptron and Random Forest, and created a dictionary to store all the models in one place, once the CT-GAN produces the synthetic data, we plot a graph to compare the real data and synthetic data. Once the model classifiers are trained on this balanced data, we will test these classifiers on the original data. Finally, at the last stage the performance of all models is evaluated using metrics such as the AUPRC curve.

5 Implementation

5.1 Software and Technologies Used

These are the system tools and libraries needed to conduct this study.

- **Programming Language Used:** Python

- **IDE:** Google Collab
- **Database:** Google Drive
- **Python Libraries/Modules**
    - For Data Understanding: NumPy, Pandas
    - For Generating Synthetic data through CT-GAN: tensorflow,pygan
    - For Data Visualization: matplotlib, plotly, seaborn
    - For Modelling: sklearn

5.2 Implementation Steps

This section provides a thorough explanation of the implementation procedures used in this study to produce the desired outcomes.

Data Acquisition Step

The CSV-formatted dataset used in this research comprises credit card transactions from European cardholders of the year 2013. There are 31 total variables in the dataset, all of which are coded in form of PCA components to ensure that no sensitive data is impacted. The dataset was downloaded from the Kaggle repository, and the CSV file was then saved to Google Drive. By granting the request made by Google Collab, the data is accessed. We next begin by comprehending the dataset, including its overall size, the number of columns, and the data types of each column.

Data Pre-processing or Exploration Step

In the data exploration step, we discovered that the dataset do not have any missing values however there were some duplicate values. Thus, these duplicate values were eliminated from the dataset. Additionally, we discovered that this dataset is significantly unbalanced after calculating the percentage of fraudulent and legitimate transactions, which was 0.173% for fraudulent transactions and 99.827% for legitimate transactions. To eliminate the outliers, two variables—Time and Amount—are scaled. The highly skewed dataset is then displayed through various dataset visualization such as pie charts, and graphs of each variable. Using scatterplots between fraud transactions and financial data, we also attempt to identify any relationships between different data factors. We discovered some findings such as :

- Fraud transactions typically do not exceed $2500 in value.
- Additionally, Fraudulent transactions are evenly spaced throughout time.

Apply CT-GAN to generate Synthetic Data

As the dataset is highly skewed, so in order to address this problem, we employed the unsupervised Conditional GAN technique, which creates synthetic data. An open-source Python package is used by GAN. First, we declare the discriminator that uses the feature matrix as input and forecasts if anything is deceitful or real. We combine the two layers of latent space and latent vector which act as the input layer, as well as build two hidden layers in between along with one output layer, in the discriminator class, and then build a model using this structure. A similar process is done with the generator. Now that discriminator and generator have been integrated into CGAN, the discriminator is not trained so that the weights remain unchanged. After creating a CGAN model using the generator's inputs and the discriminator's outputs, the model is then assembled with a learning rate of 0.0002. To make the fraudulent data appear at least significant, it is oversampled and data frames are shuffled.

Model training is done with epochs of 10. To compare the actual and false sample data, a scatter plot is made. To determine the efficiency of CGAN, we have performed training on synthetic data and performed testing on actual data.

Modeling and Implementation Steps
To determine which model performs the best in the dataset produced by GAN, we used three distinct models. In this study, we implemented Random Forest, Multi-layer perceptron, and Isolation Forest. To store the functions of each model, we established a common dictionary. These functions are called by passing the required parameters. Just after passing the necessary parameters, we can compare each model's precision, recall, and F1 score to determine which one best fits the situation.

Evaluation and Result
We used the Area Under the Precision-Recall Curve as the performance metric because the dataset is imbalanced and accuracy or precision would not be the best criterion. The following section describes the experiments and the evaluation results.

6 Evaluation

The below sections describe the experiments conducted in this research and illustrate how the results are evaluated.

6.1 Experiment 1: Using SMOTE Technique

SMOTE technique (Synthetic Minority Oversampling Technique) is a well-liked approach for unbalanced datasets, therefore we decided to use it in our first case study. In this method, two samples are chosen, one from the minority class and one from the nearest KNN. A new data segment is then made using these two and then added to the minority class. Multi-Layer Perceptron outperforms the other two classifier models as seen in Table 1. We only use AUC-score and Area Under the Precision-Recall Curve as evaluation criteria as the confusion matrix is not suitable for such an imbalanced dataset.

Models	AUC score
Isolation Forest	0.83
Multi-Layer Perceptron	0.72
Random Forest	0.63

Table 1: AUC Score of Models after applying SMOTE

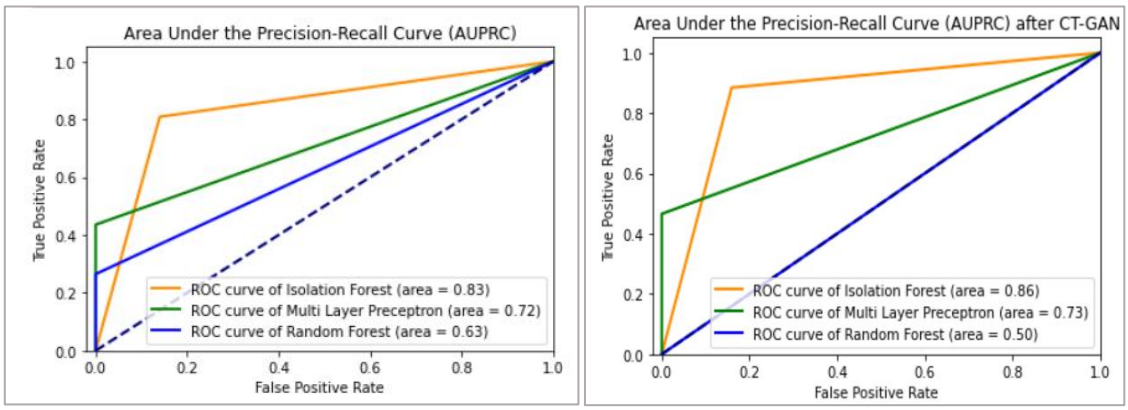
6.2 Experiment 2: Using CT-GAN

In the second experiment, we used CT-GAN. We tested the models on GAN's synthetic data and discovered that the models had considerably improved. The AUC Score has increased substantially for Isolation Forest, there is a slight increase for Multi-Layer Perceptron and a 10 % decrease for Random Forest from 0.63 to 0.53. By observing the AUC Score in Table 2, we can conclude that two out of three classifier models performed well in CT-GAN, Random forest performed well with SMOTE Technique instead of GAN, whereas the AUC Score of the other two models increased with CT-GAN.

Models	AUC score
Isolation Forest	0.86
Multi-Layer Perceptron	0.73
Random Forest	0.53

Table 2: AUC Score of Models after applying CT-GAN

6.3 Discussion



(a) AUPRC after applying SMOTE (b) AUPRC after applying CT-GAN

Figure 13

As an evaluation matrix, we have used the Area Under the Precision-Recall Curve (AUPRC) to identify the model with the highest accuracy after applying SMOTE and CT-GAN. The below curves are an illustration of the performance of all three classifiers after applying SMOTE and CT-GAN. Models with a larger area under the curve signify that they performed better. There are two out of three models where GAN is successful. Instead of GAN, the random forest performed well when using the SMOTE approach. GANs are well-liked for their simplicity in training and speed in computation. In his research, Ngwenduna(2021) also used experiments on 5 distinct datasets and found that GAN did well on 3 of the datasets, whereas SMOTE performed well on 2 of the datasets. In my research, GAN typically outperforms SMOTE on one dataset while performing well on two. Overall, SMOTE and GAN each take a different approach to addressing the problem of class inequality and each does so in a distinctive manner.
The research by Wenzel, M. (2022) illustrating different types of GAN models and their limitations also states that although GAN has advanced over time and continues to evolve with various types of GAN such as CT-GAN that attempt to overcome major challenges of GAN, it still has significant data difficulties issues such as non-convergence, instability, and noise influenced decisions, which may be the reason for the low AUPRC score for one classifier model.

7 Conclusion and Future Work

Imbalance datasets remain a major challenge for credit card datasets as briefly discussed in the literature review. In this study, we employed two different approaches to address this

major challenge of class imbalance: SMOTE technique and CT-GAN, and used three different model classifiers. Considering the AUPRC curve, we can conclude that the Isolation Forest is a more effective technique out of the three different models in both the cases of SMOTE and CT-GAN, detecting 86% of credit card fraud correctly. In future work, to improve the performance more, we can try adjusting various other parameters of the models, such as the learning rate, node count, or estimator number, to improve the results. Furthermore, as shown in Figure 13, CT-GAN assisted in improving the overall results and in the future can be used in a variety of other research domains.

References

A.Aggarwal, M. Mittal, G. Battineni(2021): Generative adversarial network: An overview of theory and applications.

Benchaji, I., Douzi, S., El Ouahidi, B. et al (2021). Enhanced credit card fraud detection based on attention mechanism and LSTM deep model. *J Big Data 8, 15.*

Brennan, P. (2012). A comprehensive survey of methods for overcoming the class imbalance problem in fraud detection, *Institute of technology Blanchardstown Dublin, Ireland*.

Dr. K. Kaur(2021): Credit Card Fraud Detection using Imbalance Resampling Method with Feature Selection.

Forough J.,Momtazi(2021).Sequential credit card fraud detection: A deep neural network and probabilistic graphical model approach.

Lin, T., Ruey J.(2021).Credit Card Fraud Detection with Autoencoder and Probabilistic Random Forest.

Gui, J., Sun, Z., Wen, Y., Tao, D. and Ye, J. (2020). A review on generative adversarial networks: Algorithms, theory, and applications.

Guzmán-Ponce, A., Sánchez, J., Valdovinos, R. and Marcial-Romero, J.(2021). DBIG-US: A two-stage under-sampling algorithm to face the class imbalance problem. Expert Systems with Applications, *168, p.114301*.

H. Liu , H. Zhao , J. Wang , S. Yuan , and W. Fen(2022): LSTM-GAN-AE: A Promising Approach for Fault Diagnosis in Machine Health Monitoring.

Khatri, S., Arora, A. and Agrawal, A. P. (2020). Supervised machine learning algorithms for credit card fraud detection: a comparison, *pp. 680–683*.

Negi S., S. K. Das and R. Bodh. (2022).Credit Card Fraud Detection using Deep and Machine Learning. *International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2022, pp. 455-461.*

Ngwenduna, K. S. and Mbuvha, R. (2021).Alleviating class imbalance in actuarial applications using generative adversarial networks, *Risks 9(3): 49.*

Roseline, F., J., Naidu, G.B.S.R., S. Pandi, V., Al. Rajasree, S., Mageswari, D.N.(2022). Autonomous credit card fraud detection using machine learning approach. *Computers and Electrical Engineering*, 102, art. no. 108132.

Shanmugapriya, P., Shupraja, R. and Madhumitha, V.(2022) Credit Card Fraud Detection System Using CNN.

S. Makki, Z. Assaghir, Y. Taher R. Haque, M. Hacid, H. Zeineddine(2019). An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection.

Tanouz D. ,Subramanian R., D. Eswar, G. V. P. Reddy, A. R. Kumar, C. Praneeth(2021). Credit Card Fraud Detection Using Machine Learning. *International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 967-972.*

Wang, D., Lin, J., Cui, P., Jia, Q., Wang, Z., Fang, Y., Yu, Q., Zhou, J., Yang, S. and Qi, Y. (2019). A semi-supervised graph attentive network for financial fraud detection, *pp. 598–607*.

Wenzel, M.(2022). Generative Adversarial Networks and Other Generative Models. *arXiv preprint arXiv:2207.03887*.

Yilmaz, I., Masum, R. and Siraj, A. (2020). Addressing imbalanced data problem with generative adversarial network for intrusion detection, *pp. 25–30*.

Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y Pan. (2021). Generative Adversarial Networks: A Survey Toward Private and Secure Applications. : ACM Computing Surveys: *Vol 54, No 6. 2022. ACM Computing Surveys*.

Zadafiya, N., Karasariya, J., Kanani, P. and Nayak, A. (2022). Detecting Credit Card Frauds Using Isolation Forest and Local Outlier Factor. *4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2022, pp. 1588-1594.

Zhang, Z. and Huang, S. (2020). Credit card fraud detection via deep learning method using data balance tools, *2020 International Conference on Computer Science and Management Technology (ICCSMT), pp. 133–137*.