***Navigating through financial challenges*** ***by harnessing the power of synthetic data***

A close-up of a logo

Description automatically generated

Master of Science (MS) in Data Science

Module: ITC6421B1 – Capstone Project

Instructor: Dr. Elena Chatzimichali

Students’ names & IDs:

Alkiviadis Kariotis 241735

Evangelos Tsilikas 271396

Konstantinos Kogias 271548

Stamatios Lykos 273820  
 

Term: Spring Term 2024

Type: Business - Report

Submission Date: Wednesday 24th of July

Word Count: 10528

Table of Contents

[Abstract 3](#_Toc171851012)

[Keywords 4](#_Toc171851013)

[Introduction 4](#_Toc171851014)

[Background on Time-series Data 4](#_Toc171851015)

[Purpose and Scope of the Project 5](#_Toc171851016)

[Literature Review 6](#_Toc171851017)

[*Overview of Synthetic Data* 6](#_Toc171851018)

[Need and importance for Synthetic Data in Various Domains 6](#_Toc171851019)

[Use of Synthetic Data in Finance 8](#_Toc171851020)

[Synthetic Data Generation 9](#_Toc171851021)

[Data Generation vs. Data Augmentation 9](#_Toc171851022)

[Discriminative vs. Generative Models 9](#_Toc171851023)

[Models and Techniques for Synthetic Data Generation 10](#_Toc171851024)

[Comparison of Techniques 12](#_Toc171851025)

[Methodology / Main Body 13](#_Toc171851026)

[Data Collection 13](#_Toc171851027)

[Sources of Data 13](#_Toc171851028)

[Description of the Dataset 13](#_Toc171851029)

[Data Preprocessing 14](#_Toc171851030)

[Data Cleaning and Normalization 14](#_Toc171851031)

[Exploratory Data Analysis (EDA) 14](#_Toc171851032)

[Data Augmentation 15](#_Toc171851033)

[Techniques Used (TimeWarp, Crop, Quantize, Drift, Reverse) 15](#_Toc171851034)

[Implementation and Results 16](#_Toc171851035)

[Model Architecture 17](#_Toc171851036)

[Overview of TimeGAN Components 17](#_Toc171851037)

[Autoencoder and its Role in Data Normalization 17](#_Toc171851038)

[Training Procedure 18](#_Toc171851039)

[Phase 1: Autoencoder Training 18](#_Toc171851040)

[Phase 2: Supervised Training 18](#_Toc171851041)

[Phase 3: Joint Training 18](#_Toc171851042)

[Synthetic Data Generation 19](#_Toc171851043)

[Process of Generating Synthetic Data 19](#_Toc171851044)

[Filtering and Enhancing Synthetic Data Quality 19](#_Toc171851045)

[Adversarial Training Process Generator Training 19](#_Toc171851046)

[Discriminator Training 20](#_Toc171851047)

[Adversarial Loop 20](#_Toc171851048)

[Model and Parameter Details 21](#_Toc171851049)

[seq\_len, n\_seq, batch\_size, hidden\_dim, num\_layers 21](#_Toc171851050)

[Training Steps and Hyperparameters (gamma, loss functions) 22](#_Toc171851051)

[Experiments and Results 22](#_Toc171851052)

[Evaluation Metrics 22](#_Toc171851053)

[Criteria for Assessing Synthetic Data Quality (PCA, t-SNE, Reconstruction Error) 22](#_Toc171851054)

[Quality of Generated Data (QoG), Privacy Preservation, Model Performance 23](#_Toc171851055)

[Description of Training and Testing Sets 23](#_Toc171851056)

[Results 23](#_Toc171851057)

[Visualization of Synthetic vs. Real Data 23](#_Toc171851058)

[Quantitative Comparison (PCA, t-SNE) 23](#_Toc171851059)

[Anomaly Detection Results 23](#_Toc171851060)

[Evaluation of Model Performance on Classification and Regression Tasks 24](#_Toc171851061)

[Discussion 24](#_Toc171851062)

[Analysis of Results 24](#_Toc171851063)

[Insights Gained from Synthetic Data Generation 24](#_Toc171851064)

[Limitations and Challenges Faced 24](#_Toc171851065)

[Conclusion 24](#_Toc171851066)

[Summary of Findings 24](#_Toc171851067)

[Implications of the Study 24](#_Toc171851068)

[Future Work and Potential Improvements 25](#_Toc171851069)

[References 25](#_Toc171851070)

[Appendices 28](#_Toc171851071)

[Appendix 1 28](#_Toc171851072)

# Abstract

In today's fast paced data-driven world, time-series data play a crucial role in various fields, such as finance, healthcare, energy etc.. However, the process of collecting high-quality time-series data is a quite difficult challenge, because of various privacy concerns, regulatory constraints, and high acquisition costs. In this report Time-series Generative Adversarial Networks (TimeGANs) are explored, for the purpose of generating synthetic financial time-series data, addressing the aforementioned challenges and constrains. The study provides a comprehensive overview of the methodologies employed during the whole journey, including data preprocessing, model architecture, and training procedures. Objectives will be covered from the theoritical foundations examination of GANs and TimeGANs, to addressing various challenges in generating synthetic financial data. Through demonstrating TimeGANs and their scalability and adaptability, across different financial contexts, the report tries to unveal their potential to revolutionize financial data analysis and decision-making processes.

# Keywords

*Time-series data, synthetic data generation, TimeGAN, financial data, machine learning,Generative Adversarial Networks (GANs)*

# Introduction

## Background on Time-series Data

In todays era, huge amounts of data have “overwhelmed” various domains. Their exististence is the main reason which has driven significant advancements in the sectors of machine learning and artificial intelligence. One category of data are time-series data, which are the recordings of observations of variables at different points in time. Such types of data are particularly important in fields such as finance, healthcare, and energy. By using time-series data we are able to capture temporal dependencies and trends. This is a crucial aspect for forecasting, anomaly detection, as well as strategic planning of an organization. For example financial time-series data, can include either stock prices, or interest rates, and economic indicators. All of the aforementioned are essential factors to be considered for decision-making in financial markets (Jansen, 2022).

However, we face challenges not only thoughout the collection of time-series data but also in their use. In many cases, obtaining comprehensive and high-quality datasets can be a very difficult procees due to reasons such as privacy concerns and regulatory constraints, in combination also with the high cost of data acquisition. These issues are especially pronounced in sectors like finance and healthcare, where those kind of data must be carefully protected in order to ensure compliance with the regulations such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA) (Financial Conduct Authority, 2020; JPMorgan, 2023).

Because of the dynamic nature of the financial markets that are in interest to us, this ads another layer of complexity to the handling of time-series data. Market conditions in this idustry tend to rapidly change due to geopolitical events, economic policies, and other factors, leading that way to volatile data patterns. This volatility necessitates robust data analysis methods in order to be able to adapt to changing conditions and provide that way accurate forecasts and insights. Advanced techniques such as Generative Adversarial Networks (GANs) and more specifically TimeGANs that are going to be studied, have been able to emerge as powerful tools for generating synthetic financial time-series data because not only they reflect these dynamic patterns, but at the same time they are ensuring data privacy and compliance (Goodfellow et al., 2014; Yoon et al., 2019).

Moreover, financial time-series data come in high-frequency, such as minute-by-minute stock prices or transaction records. This opens up another significant challenge in terms of storage, meaning where those large amounts of data will be stored, and then processed, and finally analysed. Sophisticated algorithms and high comuptational power are some tools that can help us handle such large volumes of data. So, techniques like TimeGANs, are not only able to aid towards managing this complexity but they can also provide scalable solutions, that they could be applied across different financial instruments and markets. This scalability is as we aforementioned essential for financial institutions, which are dealing and handling big data, which they analyze and use the resutls that they obtain for real-time decision-making and strategic planning (Heka AI, 2023; Towards AI, 2023).

## Purpose and Scope of the Project

With this report we aim to provide a comprehensive overview of generating synthetic data with the use of TimeGANs, and we will thorougly describe the methodologies that we used, the experiments that we did, and finally the results that we obtained. More specifically the objectives of this project are:

1. **Exploring the Theoretical Foundations:** We start from diving into the theoretical background of GANs and TimeGANs, in order to examine not only how these models function but also their principles that underly their operation (Goodfellow et al., 2014; Yoon et al., 2019).
2. **Addressing Specific Challenges**: The process of generating synthetic financial time-series data can have many challenges such as ensuring data privacy, handling data scarcity, and at the same time maintaining the temporal coherence of the generated sequences. Through this report will explore these challenges and the solutions that we are able to provided through the use of TimeGANs (Financial Conduct Authority, 2020; JPMorgan, 2023).
3. **Detailed Experimental Setup and Results**: We will describe the whole process from the beginning to the end, starting from the data preprocessing steps, the architecture of the models usedf, the experimental setup used to train and evaluate the TimeGAN models and finally the whole training and testing procedures. The results section will contain all the outcomes of these experiments, including not only visualizations but also quantitative evaluations of the generated synthetic data that we are able to produce (Yoon et al., 2019; Towards Data Science, 2023).
4. **Assessing Practical Applications:** Finally, in the discussion section we will analyze the practical applications that synthetic financial time-series data can be used for, including their use in sectors like risk management, fraud detection, and compliance with privacy regulations. We will try to highlight how synthetic data can enhance these applications and maybe even drive innovation in the financial sector (Financial Conduct Authority, 2020; JPMorgan, 2023; Towards AI, 2023).

In summary, with this report we will try to explore not only the technical aspects of TimeGANs but also to highlight their transformative potential that they can have in the financial industry. Through providing a detailed analysis of the methodologies that we used, the challenges that we faced, and the practical applications on which we were able to test our models, we seek to demonstrate how synthetic time-series financial data are able to revolutionize not only data analysis techniques but also the decision-making process. Finally, we also aim to show the potential that TimeGANs have in terms of scalability and adaptibility, across different finincial contexts, by illustrating their versatility in the generation of synthetic data, that can be used to enhance diverse applications.

# Literature Review

#### Overview of Synthetic Data

### Need and importance for Synthetic Data in Various Domains

When it comes to obtaining data and anlysing them several concers arise. These concerns have to do with privacy preservation, data scarcity, cost efficiency, and collaboration issues either between different organizations or even internally in an orgmanization because of the data silos. In order to avoid all those constraints we have turned our attention to synthetic data generation. This offers a solution to many of these challenges. Synthetic data are artificially generated data, who mimics not only the statistical properties of the real-world data that they have been used in order to be generated but also their patters and anomalies. One very useful situation that they can be used is for augmenting existing datasets, allowing that way for more robust and comprehensive analyses. The generation of synthetic data has several key benefits concering:

1. **Privacy Preservation**: Through creating synthetic versions of the original and most probably sensitive datasets, organizations are able to share and analyze data without worrying about individual privacy, because of the fact that synthetic generated data are anonymized datasets that mimic the properties of the real ones. This way they can be aligned with the various compliances regarding privacy regulations and protecting of personal information, in fields like finance and healthcare where data privacy is a paramount concern (Confident AI, 2023). So, synthetic data provides a way to balance the nowadays necessary need for data access with the imperative to protect privacy according to all the regulations such as GDPR and HIPAA which have set very strict rules towards sharing personal data (Financial Conduct Authority, 2020).
2. **Addressing Data Scarcity**: In the business sectors where collecting large amounts of real-world data can be challenging, difficult, very expensive, or even infeasible in certain fields, synthetic data provides tries to eliminate those barriers through a practical alternative. By generating additional synthetic data points, that reflect and follow the properties of the original dataset, synthetic data opens the way for the development of machine learning models with better accuracy and reliability (Guim Perarnau, 2017; Towards AI, 2023). In scenarios like autonomous driving where it is difficult to collect data for all rare events and cover all the possible gaps, this could be proved very usefull when acritical driving scenario appears which may not be well-represented in the collected original data, but generated in the synthetic (Medium - Microsoft, 2023a; Forbes, 2024).
3. **Cost Efficiency**: The traditional process of data collection requires a significant amount of time as well as resources for the data gathering process, the cleaning of the data, and their annotation. On the other hand, synthetic data generation models such as GANs, are able to quickly produce large volumes of data with the minimal manual effort. This level of efficiency that they provide, makes it easier to focus on model development and testing rather than collecting etc., and ultimately leading to accelerated innovation and reduced costs (Forbes, 2024).
4. **Enhanced Model Training**: Concerning machine learning models, the quantity and diversity of the training data plays the most crucial role for achieving high performance in the models. Synthetic data have the ability to significantly enhance the training process of the machine learning models, by being able to provide large, rich and diverse datasets, that will cover a wide range of different scenarios. As we have aforementioned, this can be very handy when it comes to the training of models in environments or industries where real-world data are limited or even difficult to obtain (Towards AI, 2023).
5. **Testing and Validation**: Synthetic data can be used towards testing and validation of the developed machine learning models. If the generated synthetic data include edge cases and rare events, it will enable researchers to be more confident that their models are robust and that they are able to handle a wide range of situations. This can again apply in industries where we have occurences of critical or unkown application or events such as autonomous driving and financial risk management, where failure to handle rare events can cause serious consequences (Financial Conduct Authority, 2020).
6. **Facilitating Collaboration**: In many industries or even better internal departments within an organization, collaboration can be proven essential for advancing research and development. However, data sharing between organizations is often hindered by privacy concerns and regulatory constraints, and internally data silos are causing departments not to share data. Synthetic data can provide a solution to this situation, by allowing organizations to share data without compromising privacy. This way not only collaboration can be boosted but also sharing of valuable insights and knowledge can be possible between the different parties that are involved in the specific process (Confident AI, 2023).

### Use of Synthetic Data in Finance

#### Regulatory Compliance and Privacy

When it comes to the financial sector, data privacy and regulatory compliance are two factors that are of utmost importance. Financial institutions tend to handle vast amounts of sensitive information, which may include transaction records, customer details, or even financial statements. So, ensuring that these data are secured in terms of the privacy, is very critical in order to maintain customer trust and at the same time comply with regulations such as GDPR and the Gramm-Leach-Bliley Act (Financial Conduct Authority, 2020). Synthetic data are able to provide a solution to this situation, through the creation of anonymized datasets that will retain the statistical properties of real data without exposing any sensitive information. That benefit, will allow financial institutions to perform their analysis and develop their models while at the same time adhering to privacy regulations (JPMorgan, 2023). One example usage of synthetic data in the financial sector could be to train machine learning models for credit scoring, fraud detection, and risk assessment and at the same time adhering to the privacy of individuals. Through the use of synthetic data, financial institutions have the ability to share datasets with external researchers and partners, opening up the path for further research and innovation possibilities (Towards Data Science, 2023).

#### Risk Management

Another place where synthetic data can be found usefull in the financial sector, ir the risk management. In their daily basis, many financial institutions must assess and mitigate risks which could be associated with various market conditions, or economic scenarios, and maybe operational factors. In order to formulate their stategies and their decision making processes, they will need to simulate hypothetical scenarios, in which synthetic data can be proved helpful. If we look for example a bank case, a dataset could be created that will mimic economic downturns or even market crashes, enabling the bank that way to enhance their strategies and rely their decisions not only on historical data but also on the potential future events that have been produced through the synthetic data. This is one reason that showcases how they could develop a more robust risk management frameworks that can withstand adverse market conditions (Financial Conduct Authority, 2020; JPMorgan, 2023).

Additionally, synthetic data can be also used to simulate rare events that maybe will have a severe impact such as a financial crises or systemic shocks. Even though this scenarios that pose significant risks, can be present in the historical data, through leveraging synthetic data they will be able to enhance their models robustness and their overall strategy (JPMorgan, 2023).

#### Fraud Detection

Fraud detection is another critical application in the finance sector where synthetic data could aid. The detection of fraudulent activites are in the daily basis of financial institutions, and it keep evolving into new patters frequently. Synthetic data can help financial institutions adress this fast changing pace patterns, through generating diverse and realistic fraudulent activity patterns, that have not being seen before into the historical data. This will also enhance the training of their machine learning models, and make them more robust and effective in identifying and preventing fraud, enabling them to stay ahead of sophisticated fraud scheme, anticipate and respond to new fraud tactics faster and at the same time protect their financial assets and their customer information (Towards Data Science, 2023).

## Synthetic Data Generation

### Data Generation vs. Data Augmentation

Data generation is the process of creating entirely new data points that mimic the properties of the real data. This process is typically done using models like GANs, VAEs, and Large Language Models (LLMs). That generated data can be used either to supplement existing datasets or to create entirely new datasets. For example, GANs are able to generate realistic images that could be used towards training computer vision models. On the other hand, VAEs can generate diverse synthetic data for various applications (Perarnau, 2017, Sharifi, n.d). On the other hand data augmentation, is the process of creating new data points by modifying the existing data. This process, is focused on the enhancement of the diversity of the training dataset, through applying transformations such as rotation, flipping, cropping, and adding noise to images, or if needed generating new samples to deal with imbalances between classes, by oversampling the minority classes in the dataset. It is a technique which is widely used towards the improvement of the robustness and the performance of machine learning models. In image classification tasks example, augmented data will aid models to generalize better, because they will be exposed to a wider variety of samples during the training phase (Tan, 2021; XQ,n.d.).

### Discriminative vs. Generative Models

#### Definitions and Differences

Artificial Intelligence (AI) models can be broadly classified into two large categories: discriminative models and generative models. Discriminative models, such as logistic regression, Random Forests (RF), and Support Vector Machines (SVM), are models that are designed to classify input data into predefined classes. They model the conditional probability of the target variable given the input data (P(y|x)) and they are mostly used in applications for spam detection, image recognition, and medical diagnosis (Guim Perarnau, 2017; Towards Data Science, 2023).

On the other hand, generative models aim towards understanding and modeling the entire distribution of the input data. So, through their learning from the joint probability distribution (p(x,y)), generative models are able to generate new data points that resemble the original data points. Examples of generative models are Latent Dirichlet Allocation (LDA), Variational Autoencoders (VAE), and Generative Adversarial Networks (GANs). GANs are generative models that are mostly notable for their ability of generating highly realistic images, videos, and text (Perarnau, 2017).

The main difference, of the two aforementioned types of models can be found in their objectives. Generative models, are used to generate new data point that mimic the originals data set ditribution and discriminative models are models that are used for discriminative purpose in classification tasks (Towards AI, 2023; Heka AI, 2023).

#### Applications and Examples

Application examples of discriminative models are in spam detection, image recognition, and medical diagnosis, and in general in the cases that the goal is to categorize data accurately. These models are performing well when it comes at finding decision boundaries that separate the different classes in the data set. For example, logistic regression is used for the prediction of the probability for an input to belong to a particular class, and SVMs find the hyperplane that it is the best choice for the separation of the different classes in the feature space (Towards AI, 2023; Medium - Microsoft, 2023b).

Generative models, on the other hand, are more applicable when it comes to data augmentation, anomaly detection, and more. For instance, in image generation scenario, GANs are able to create new images that can’t be easily distinguished from real ones. This results can be used towards training datasets for computer vision tasks, with the augmented images that have been produced from GANs. In anomaly detection scenario, there VAEs can learn the distribution of normal data and therefore being able to identify anomalies as deviations from this learned distribution from the real data (Towards Data Science, 2023). Also, another field that generative models play a crucial role is in natural language processing (NLP). Models like a version of chatGPT (GPT- 4 for example) can generate coherent and contextually relevant text based on specific input. Finally, it can also be applicabel in healthcare sector, where they can be used towards synthesizing medical records, enabling that way the conduct of studies without compromising patient privacy (Confident AI, 2023).

### Models and Techniques for Synthetic Data Generation

Throughout the years several techniques have been developed for synthetic data generation, each with its advantages and limitations:

* SMOTE and ADASYN: These techniques were developed towards addressing class imbalance in datasets. This was done through the generation of synthetic samples for the minority class. SMOTE (Synthetic Minority Over-sampling Technique) is a model that creates new synthetic instances by interpolating between existing minority class examples, while ADASYN (Adaptive Synthetic Sampling) is a model tha focuses more on generating synthetic samples for the minority class examples that are a bit harder to classify. Both techniques have been extensivlybeing used in applications such as fraud detection and medical diagnosis, where they were implemented in order to improve the performance of classification models on imbalanced datasets (Towards Data Science, 2023; Medium - Microsoft, 2023a).
* Variational Autoencoders **(VAEs):** VAEs belong into the generative model damily, and are models that learn the distribution of input data and generate new data instances by sampling, using the knowledge that they have acquired from the distribution that they have seen. They consist of two parts, one is an encode, which is responsible for compressing the input data into a lower-dimensional latent space and the second part is a decoder, who is respponsible for reconstructing the data from the latent representation. VAEs have found usage in applications such as image generation, anomaly detection, and data augmentation, and while they are effective at capturing the overall data distribution, it is possible that VAEs would generate blurrier and less detailed outputs compared to GANs (Medium - Microsoft, 2023a; Towards AI, 2023).
* Generative Adversarial Networks **(GANs)**: GANs are the models that have revolutionized the field of synthetic data generation. They consist of two neural networks: the first one is a generator, who is responsible for creating synthetic data and other neural network is the discriminator, who is resbonsible for checking the generated synthetic data from the discriminator in terms of evaluating their realism. The training proccess between the two networks can be charectarized as adversarial, and it helps the generator towards producing high-quality synthetic data that mimics as close as possible the real data. These types of models have found successful applications in various domains, including image synthesis, video generation, and time-series data generation (Goodfellow et al., 2014; Yoon et al., 2019; Towards AI, 2023).

### Overview of Generative Adversarial Networks (GANs) & TimeGANs

Generative Adversarial Networks (GANs), which were introduced by Ian Goodfellow and his colleagues back in 2014, they represent a breakthrough in the field of artificial intelligence (Goodfellow et al., 2014). GANs are nueral networks that consist mainly of two parts: a generator and a discriminator, which are simultaneously trained through a process of adversarial learning. The generator is the one responsible for the creation of the synthetic data, while the discriminator is the ine responsible for their evaluation, in terms of their realism, by comparing the synthetic data to the real data. Through this competitive interaction procees, both networks are driven towards continuous improvement, which will finnaly result in generating highly realistic synthetic data (Goodfellow et al., 2014; Towards Data Science, 2023).

The generator starts the whole process by adding random noise in the real data and then transforming them into synthetic data. The discriminator, on the other hand, is the one who attempts to distinguish between real and synthetic data. Through each iteration of the training progresses, the generator becomes better at producing realistic data, and at the same time the discriminator becomes better at identifying synthetic data. The goal of this neural network, is to reach a point where the discriminator will no longer be able to reliably distinguish between the real and synthetic data (Yoon et al., 2019).

A screenshot of a computer

Description automatically generated

Figure 1: Overall Architecture of GANs (Mohammadi,2021)

GANs have already been successfully applied into various domains, which they include text generation or image and video synthesis. The most recent developments is the one of Time-series Generative Adversarial Networks (TimeGANs), which represents a significant advancement, because of the fact that these models are specifically designed towards handling the unique challenges in the procedure of generating time-series data, like being able to capture temporal dependencies and at the same time maintain the coherence of generated sequences (Yoon et al., 2019; Towards AI, 2023).

TimeGANs use both supervised and unsupervised learning objectives and techniques, being able that way to combine the benefits of GANs together with the ability to capture temporal dynamics in the real time-series data, towards learning a time-series, through incorporating it’s autoregressive nature (Tan, 2021; Sharifi, n.d.). This makes them a very well-suited choice for applications in sectors like finance, healthcare, and other domains where time-series data is prevalent (Yoon et al., 2019; Heka AI, 2023). The main difference between the two, is that GANs are more effective for tasks where the temporal component is not a factor, like image synthesis or video generation for example, but on the other hand TimeGANs are specifically design for generating time-series data, being able at the same time not only to capture recurrent components but also to temporal dependencies(Medium - Microsoft, 2023a; Rochetti, De Gregorio, Basile, & Osmani, 2024).

### Comparison of Techniques

The table below summarizes the advantages and also the limitations, of the key techniques used for synthetic data generation:

|  |  |  |
| --- | --- | --- |
| **Technique** | **Advantages** | **Limitations** |
| **GANs** | 🡪High-quality data generation  🡪 Able to capture complex distributions | Mode collapse, difficult to train, requires large datasets (Goodfellow et al., 2014; Heka AI, 2023) |
| **VAEs** | 🡪Captures overall data distribution well  🡪Are easier to train | Generates blurrier, less realistic data in comparison to GANs (Towards Data Science, 2023) |
| **SMOTE/ADASYN** | 🡪Simple to implement  🡪 Effective for class balancing | Limited to tabular data 🡪 which can lead to overfitting if not used properly (Towards Data Science, 2023) |

In conclusion, GANs and TimeGANs offer powerful tools for generating synthetic data, particularly in finance sector, being very good at generating realistic synthetic data, by being able to capture all the complex data distributions, even though they are challenging to train and they are capable of experiencing mode collapse. VAEs are easier to train but may produce less detailed outputs. SMOTE and ADASYN balance class distributions in tabular data but at the same time they have limited applicability to other data types. And now Diffusion models are starting to enter the scene and gaining much attention because of their detailed data structures. They iteratively add and reverse noise to create high-fidelity synthetic data (Rochetti, De Gregorio, Basile, & Osmani, 2024).

# Methodology / Main Body

## Data Collection

### Sources of Data

The data that we used for our project were acquired from the Federal Reserve Economic Data (FRED) API, which is a source that mainly offers economic and financial data sets. Through FRED we were able to gain access to a numerous amount of time-series datasets, from multiple and different sources, which may be originated either from government agencies, research organizations, or even international bodies. For this project, we focused mainly on the daily price series of crude oil benchmarks - West Texas Intermediate (WTI) and Brent Crude. These data are well-documented historical data, which spans from May 20 of 1987 until to December 31 2020.

Description of the Dataset  
  
So, the dataset comprises of daily closing prices for WTI and Brent Crude oil, of approximately 33 years. The prices that are depicted in the dataset, reveal critical indicators of economic activity and market sentiment, which makes them ideal towards studying the effectiveness that synthetic data generation models can provide in such cases. The original dataset came with missing values, so it needed to be cleaned in order to ensure consistency. The dataset was lenghty and had breadth and because of that it provided a good foundation that we could train our TimeGAN model on. It gave us the opportunity to train and “force” our model towards learning patterns and trends that are inherent in the original financial time-series data set that we had.

## Data Preprocessing

### Data Cleaning and Normalization

Data nowadays are produced constantly and in large numbers, and because many times they are automatically produced through softwares that may face some incosistencies, this fact makes data cleaning a critical and necessary step in order to ensure not only the dataset's quality but also it’s reliability. So, our process of data cleaning began firstly by identifying and then removing any rows which had missing values, in order to maintain the integrity of the original time-series data. So, because of the importance of scaling in neural network training process, we proceeded with normalizing the data, through the use of the MinMaxScaler, which scaled all features to a range between 0 and 1. Through normalization we try to make sure that the model will treat all features of the dataset equally, which is a crucial aspect towards achieving a stable and efficient training, in terms of gradients. Normalization was a necessary process that needed to be done, in order to handle all the significant variations of the price variable in our dataset, which was changing from time to time in the historical data. Through applying the normalization process we were able to handle better all the different magnitudes and units of measurement across features, which ultimately allowed us for a more efficient model learning (Ali et al., 2014).

### Exploratory Data Analysis (EDA)

Then we conducted an exploratory Data Analysis (EDA), which aid us towards gaining some insights regarding our dataset's underlying structure, helping us at the same time to identify any anomalies or patterns. In order to understand our dataset’s central tendencies and variability, we computed various key statistics such as the mean, the median, the standard deviation and the correlation coefficients. More specifically correlation analysis was conducted between WTI and Brent prices, in order to explore the possible relationship of the two time-series but also dependencies between them.

The exploratory data analysis process also involved the visualization of data through various plots. Visualization is a very powerful tool, because we can easily understand quickly the behavior that our time-series data show over the time periods. We used line plots to observe the temporal trends and seasonal patterns that exist in our dataset’s prices, and more specifically of the normalized closing for WTI and Brent prices, in order to reaveal trends, seasonal patterns and potential outliers. Here is the graph:

A graph showing a line graph

Description automatically generated with medium confidence

Figure 2: Normalized Closing Price for WTI

Furthermore, we created some heatmaps in order to view better the correlation between the different variables in our dataset, gaining that way a more clear picture of how the prices of WTI and Brent are related. Here is the plot:

A blue squares with white text

Description automatically generated

Figure 3:Correlation between WTI & BRENT

By applying all the aforemention visualizations, we are able to confirm visually that the data preprocessing steps that were applied previously, were towards the right direction and that we are ready for an effective model training, revealing us at the same time valuable insights that would be used afterwards for the training of our TimeGAN model in order to capture its relationships more efficiently.

## Data Augmentation

### Techniques Used (TimeWarp, Crop, Quantize, Drift, Reverse)

In order to enhance the diversity of the training dataset that we used, we applied several data augmentation techniques. These techniques resulted in the creation of new data points, by altering the original time-series data in various ways, enhancing our model’s ability to generalize better because it has been exposed to a broader range of variation throughout it’s training (Nikitin, 2023). The techniques that we used are:

* **TimeWarp**: This technique is used to distort the time axis of the original data, adding variability in their temporal dynamics without altering at the same time the overall trend.
* **Crop**: By cropping randomly, we selected subsections of our data, ensuring that way that our model will be exposed to different segments of our original time series.
* **Quantize**: This method was used in order to reduce the precision of the data. This was achieved through the creation of a more simplified version of the time series, which will retain at the same time all the essential characteristics.
* **Drift**: Drift is a technique that introduces a small but consistent change over time, which is able to simulate the gradual trends or shifts in the data.
* **Reverse**: This technique is used to flip the time-series data, and it provides a different perspective and at the same time it helps our model to see and learn symmetrical patterns.

### Implementation and Results

So, after the augmentation process was completed, we compared the augmented data to the original one’s, in order to verify that we maintained all the essential characteristics of the dataset while at the same time introducing variability. For the comparison process, we took advantage of both the various aforementioned visualizations as well as the statistical measurements, towards verifying that the techniques applied successfully increased the dataset's diversity while at the same time retaining all the key features and temporal dynamics of the original dataset. And here are some plots:

A graph showing the difference between bitcoin close prices

Description automatically generated

Figure 4: Comparison of Original & Augmented Data (with Crop, Quantize & Drift)

A graph showing the price of bitcoin

Description automatically generated

Figure 5: Comparison of Original & Augmented Data (with Reverse, Crop, Quantize & Drift)

## Model Architecture

### Overview of TimeGAN Components

So, afterwards we started to set up the TimeGAN model. This model consists of several key components, which are designed to work together in a cohesive manner in order to generate realistic synthetic time-series data, through leveraging both supervised and adversarial learning techniques at the same time, towards generating high-quality synthetic time-series data (Xu, Skoularidou, Cuesta-Infante, & Veeramachaneni, 2019). The components of the model are:

* **Embedder**: The embedder network is responsible for compressing the input data into a lower-dimensional latent space and at the same time capturing essential features and patterns in the time series.
* **Recovery**: The recovery network is responsible for the reconstruction of the original data from the latent space representation, ensuring at the same time that the latent features will be able to br accurately decoded back into the original data format.
* **Generator**: The generator network is the one responsible for the creation of synthetic data, through the process of mapping random noise vectors to the latent space, aiming to produce data that resembles the one’s of the real dataset.
* **Discriminator**: The discriminator network is the one responsible for the evaluation of the authenticity of the generated data, which is achieved through the distinguish between real and synthetic data points. At the same time the discriminator provides feedback from its execution to the generator, so that the generator can improve its performance.
* **Supervisor**: The supervisor is an (optional) network, that aids the generator through the prediction of future latent representations, ensuring that way that the generated sequences will maintain temporal consistency and realistic transitions.

### Autoencoder and its Role in Data Normalization

The Autoencoder plays a crucial role in the data normalization process and it is used in order to learn a compact representation of the data and also ensuring that the data are represented in a lower-dimensional space, resulting into facilitating a more efficient and easier training of the TimeGAN model, ensuring at the same time that the generated data closely match the real data in terms of distribution and temporal dynamics. An autoencoder is comprised by an encoder and a decoder. The encoder is responsible for compressing the input data into a latent space and capturing at the same time all its essential features, and adterwards the decoder takes over, who reconstructs the original data from this latent representation (Jiang, Kim, Guan, & Le, 2018).

## Training Procedure

### Phase 1: Autoencoder Training

The first phase of training procedure is about the training of the autoencoder, towards learning an efficient representation of the input data. As we aforementioned, the encoder compresses the data into a lower-dimensional latent space, and then the decoder reconstructs the data from this representation. This phase not only helps towards capturing the essential features of the time-series data but also at ensuring that the subsequent training phases will be able to build upon a robust representation. So, during this phase the autoencoder is trained using the mean squared error (MSE) loss function, which is a measure that is used to measure the difference between the original data set and their reconstruction, having as goal the minimization of this reconstruction error, ensuring at the same time that autoencoder can accurately encode and decode the time-series data (Xu, Skoularidou, Cuesta-Infante, & Veeramachaneni, 2022).

### Phase 2: Supervised Training

In the second phase or in the supervised training phase, we ensure that model learns the temporal dynamics of the data, enabling it that way to generate sequences that are not only realistic but at the same time temporally consistent. This is achieved through the training of the supervisor component, which is trained to predict future latent representations, by using supervised techniques in order to minimize the MSE loss or prediction error in the future time steps, between the predicted and the actual latent representations. This ultimately helps the model to learn all the sequential dependencies in the data, which plays a crucial role in the part of generating realistic time-series sequences (Yoon, Jarrett, & van der Schaar, 2024).

### Phase 3: Joint Training

The third and final phase involves the joint training of the generator and discriminator together, but at the same time with fine-tuning the embedder and recovery components. So, the process here is that the generator aims to produce realistic synthetic data, pass them to the discriminator and try to fool it, while at the same time the the discriminator is trying to distinguish between the real and the synthetic one’s. Throughout the specific phase, the metric that is used to measure the discriminator's ability to distinguish between real and synthetic data is asversarial loss (binary cross-entropy), for which the generator is optimized towards minimizing it and at the same time the discriminator is focused towards maximizing it. This joint training process continuously drives both of the components to improve resulting in the improvement of the quality of the synthetic data, bringing them to the stage where they can resemble as close as possible the real data, in terms of both appearance and temporal dynamics and finally producing high-quality synthetic data (Nikitin, 2023).

The Joint training and the adversarial trainnig show some differences, which are analyzed and compared in a talbe in Appendices (See Appendix 1).

## Synthetic Data Generation

### Process of Generating Synthetic Data

So, the process of generating synthetic data begins with the feeding of random noise vectors into the generator, which then transforms this to synthetic data sequences, which are then evaluated through the discriminator and then feedback is provided back to the generator towards improving its performance. This loop continues until the discriminator can no longer tell the difference between real and sythetic data, because of the refined generator’s ability that has acquired throughout this adversarial training. All this process leverages the learned latent space so as to make sure that the synthetic data that has been generated, have been able to capture all the data’s essential features and temporal dynamics.

### Filtering and Enhancing Synthetic Data Quality

In order to ensure and enhance the fact that the synthetic data resemble the real data in appearance, keeping their statistical properties and temporal dynamics but at the same time maintaining high fidelity to the real data, we applied various filtering techniques so that we can detect and remove anomalies on reconstruction errors. This process involved the use of the autoencoder in order to encode and decode the generated data by compairing the reconstruction error to a predefined threshold. The data points that were flagged with high reconstruction errors are considered anomalies and are excluded from the final synthetic dataset (Lampis, Lomurno, & Matteucci, 2023).

## Adversarial Training Process Generator Training

The generator training involves the optimization of the generator network, towards the process of producing synthetic data that are able to fool the discriminator. This whole process is achieved by the feeding of random noise vectors into the generator, from which thy synthetic data sequences are then produced. Through this iterative training process, by the help of the continuous feedback given from the discriminator, the generator aims towards minimizing the binary cross-entropy loss and therefore it becomes able at producing as much as possible realistic and high-quality synthetic data (Nikitin, 2023).

A diagram of a computer

Description automatically generated

Figure 6: Generator Training process (Mohammadi,2021)

### Discriminator Training

On the other hand, the discriminator training part focuses on distinguishing between real and synthetic data, through evaluating the data sequences and at the same time assigning them a probability score, which indicates the likelihood of each sequence being real. So, through the iterative process the objective here is to maximize the binary cross-entropy loss, ensuring that it can accurately identify and differentiate real data from synthetic data, and then feeding the results back to the generator (Nikitin, 2023).

A diagram of a process

Description automatically generated

Figure 7:Discriminator Training Process (Mohammadi,2021)

### Adversarial Loop

Moving on to the adversarial loop, which is a process that involves the simultaneous training of the generator and discriminator in a competitive manner, providing continuous feedback from one to the other. As was aforementioned the generator aims at the production of synthetic data that can fool the discriminator, while the discriminator works the ability to distinguish. This adversarial process goes on until the point where the generator can consistently produce high-quality synthetic data that the generatort cannot differentiate (Yoon, Jarrett, & van der Schaar, 2024).

A screenshot of a computer

Description automatically generated

Figure 8: Autoencoder: embedding and recovery networks Adversarial Network: sequence generator and sequence discriminator components (Autoencoder: embedding and recovery networks Adversarial Network: sequence generator and sequence discriminator (Jansen,2022)

## Model and Parameter Details

### seq\_len, n\_seq, batch\_size, hidden\_dim, num\_layers

Several key parameters were fine-tuned after several experimental tries, in order to improve the performance of the TimeGAN model (Nikitin, 2023). Here are the parameters that we used:

* **seq\_len**: The sequence length, is the number of time steps used in each data sequence, where we used a lengh of 24.
* **n\_seq**: The number of features or variables in each time sequence, which was set to 2, that represented the WTI and Brent prices.
* **batch\_size**: The number of sequences processed in each training step, which we used a batch of size of 128 as to balance training speed and memory efficiency.
* **hidden\_dim**: The number of units in each hidden layer of the neural networks, where 24 hidden dimensions were chosen, in order to capture the complexity of the time-series data.
* **num\_layers**: The number of stacked layers in the neural networks, where for most components we used three layers, in order to ensure sufficient depth and to be able that way to learn complex patterns.

### Training Steps and Hyperparameters (gamma, loss functions)

Because of the multistep training of the model, besides the previous parameters, there was also a need to tune various hyperparameters also, towards optimize model performance (Brophy, Wang, She, & Ward, 2023):

* **Training Steps**: We performed a training of 5000 total steps, with different phases focusing on training the autoencoder, supervisor, generator, and discriminator.
* **Gamma**: We used a gamma of 1, which is a hyperparameter that is used in the adversarial training process, aiding at the balance of contributions of the different loss components.
* **Loss Functions**: The Mean Squared Error (MSE) and Binary Cross-Entropy (BCE) loss functions were used to optimize the autoencoder and adversarial training, respectively. The MSE loss measures the reconstruction error, while the BCE loss evaluates the discriminator's ability to distinguish between real and synthetic data.

# Experiments and Results

## Evaluation Metrics

### Criteria for Assessing Synthetic Data Quality (PCA, t-SNE, Reconstruction Error)

The quality of the generated synthetic data was assessed using several evaluation metrics (Nikitin, 2023):

* **Principal Component Analysis (PCA)**: PCA was used to reduce the dimensionality of the data and visualize the distribution of real and synthetic data in a lower-dimensional space. By comparing the PCA plots of real and synthetic data, we assessed the similarity in their distributions.
* **t-Distributed Stochastic Neighbor Embedding (t-SNE)**: t-SNE was employed to visualize the data in a two-dimensional space, highlighting the clustering patterns of real and synthetic data. This technique helps in understanding the diversity and quality of the generated data.
* **Reconstruction Error**: Through the use of the autoencoder we computed the reconstruction error, which measures the difference between the original and reconstructed data and shows high fidelity if it is low and the opposite if the recontruction error is high.

### Quality of Generated Data (QoG), Privacy Preservation, Model Performance

Then the data were evaluated in terms of (Ashrafi, Schmitt, Spang, Möller, & Voigt-Antons, 2024):

* **Quality of Generated Data (QoG):** This was achieved through the comparison of the statistical properties and distributions between of the real and synthetic data, with metrics such as mean, standard deviation, and correlation coefficients which aided towards the evaluation of similarity between the two datasets.
* **Privacy Preservation:** In order to be able to identify whether an external observer can can distinguish between real and synthetic data points, we applied techniques like membership inference attakcs, where we tested if the sythetic data were leaking any sensitive information from the original dataset.
* **Model Performance:** Through the use of metrics such as F1 score, and Area Under the Curve (AUC), we evaluated the performance of the trained machine learning models on the synthetic data, in order to understand the level which the synthetic data helps the model to learn tasks better that the real ones.

### Description of Training and Testing Sets

To ensure that our model will be able to capture all the underlying patterns and trends effectively and at the same time to test our model on unseen data, and therefore providing an unbiased evaluation of its performance, we devided our model into training and testing sets, with a typical split of 80% for training and 20% for testing. To train the TimeGAN model we used the the training set, and then the test set to evaluate the model's performance and the quality of the generated synthetic data (Tan, Yang, Wu, Chen, & Zhao, 2021).

### Loss Functions

**Reconstruction Loss**: Confirms that the Embedder and Recovery networks can effectively rebuild the initial time-series data.  
  
A black text with black letters

Description automatically generated

**Supervised Loss**: Confirms that the Supervisor network assists the Generator create latent representations with correct temporal dynamics.

A black letter with a white background

Description automatically generated

**Adversarial Loss**: Operated for both the Generator and Discriminator to make sure that the Generator produces pragmatic latent representations.

A black line with letters and numbers

Description automatically generated

**Overall Loss**: The concluding loss function contains all the discrete losses.

A black text on a white background

Description automatically generated

#### Why Logs

**Stabilizing Training:** The log function assists in consolidating the training process of GANs by issuing **smoother gradients**. This is important for the reason that **GAN training can be blatantly unsteady as a consequence of the adversarial essence** of the Generator and Discriminator.

**Cross-Entropy Loss:** The adversarial losses in GANs are extracted from the cross-entropy loss, which measures the dissimilarity between the forecasted probability distribution and the true distribution. In binary classification (for instance discriminating real from fake data), **the cross-entropy loss as was anticipated incorporates a log term**.

## Results

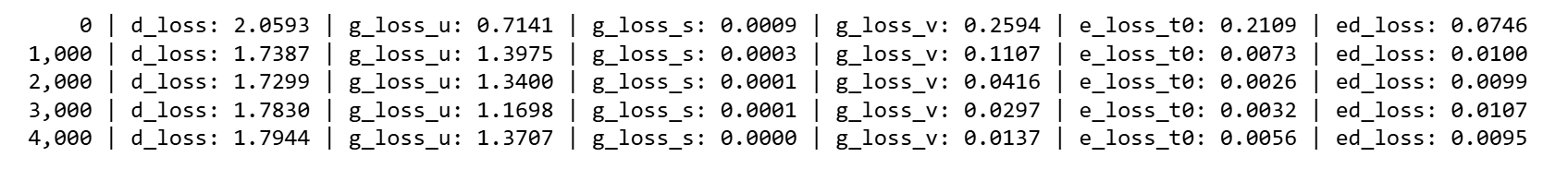


Figure 9: Loss function results every 1000 training steps.

**d\_loss** reduces initially from 2.0593 to 1.7299 but then fluctuates around 1.78, advocating a specific stability but also a likely ongoing challenge in separating real from fake data.

**g\_loss\_u** expands from 0.7141 to a peak of 1.3975 and then balances around 1.37, showing enhancing execution of the generator.

**g\_loss\_s** decreases and sticks very low, signifying solid performance in the supervised feature or minimum addition from this term.

**g\_loss\_v** decreases consistently, demonstrating increased control over the variability of the generated samples.

**e\_loss\_t0** begins fairly high at 0.2109 and falls significantly, indicating fast improvement in the encoder's execution.

**ed\_loss** remains low all through, showing superior performance in every combined task this exhibits.

In general, these logs suggest the GAN is enhancing in the long run, with both the generator and the discriminator learning and adjusting, though the discriminator indicates some variation in its loss values.

### Visualization of Synthetic vs. Real Data

A graph with blue and orange lines

Description automatically generated

Figure 10: Real VS synthetic data without excluding anomalies.

**Below you will find the analysis of a random rolling window including anomalies:**

**Trend Alignment**:

* **Initial Phase**: In the early stages (steps 0 to 2), the synthetic data (dashed orange line) indicates a steep fall followed by a prompt rise. In contrast, the real data (solid blue line) begins with an average upward trend.
* **Middle Phase**: Among steps 2 and 8, both the real and synthetic data display an upward trend, even though the synthetic data reaches its peak sooner than the real data.
* **Final Phase**: From step 8 onward, the synthetic data stabilizes at a relatively persistent level, while the real data encounters a peak around step 12 before decreasing.

**Deviation Analysis**:

* **Initial Deviation**: There is a significant difference between the synthetic and real data in the first steps (0 to 2), indicating that the synthetic model might initially be less precise or in a learning stage.
* **Convergence**: After the initial phase, the synthetic data appears to follow a similar overall trend to the real data, with both reaching their individual peaks around the middle steps.
* **Stabilization vs. Variability**: The synthetic data stabilizes more rapidly and persists relatively flat, while the real data continues to express more natural variability, peaking and then declining. This illustrates that the synthetic model perhaps is smoothing out some of the natural fluctuations seen in the real data.

**Overall Evaluation:**

* **Strengths**:
  + **General Trend**: The synthetic data expresses the overall upward and then stabilizing trend seen in the real data, showing that the model has acquired the knowledge of the general pattern.
  + **Stabilization**: The synthetic data indicates satisfactory stability after the initial learning phase, which is a useful sign of convergence and reliability of the synthetic model.
* **Weaknesses**:
  + **Initial Instability**: The initial sudden changes in the synthetic data encourage that the model may have an unstable learning phase or may be overly reactive to initial conditions.
  + **Lack of Variability**: The synthetic data seems to be less variable than the real data, probably indicating that the model is not capturing all the shades of the real data's behavior.

We cannot make final evaluation though the depiction of random rolling windows, but it is useful to help us understand how the synthetic data behave contrary to the real data.

Below you will find another random rolling window with totally different graphic depiction, but still the real and synthetic data move similarly with each other depending on the case.

A graph showing the value of a graph

Description automatically generated with medium confidence

### Quantitative Comparison (PCA, t-SNE)

A screenshot of a diagram

Description automatically generated

Figure 11: Assessing Diversity with PCA and t-SNE to real VS Synthetic data

**PCA (Principal Component Analysis) Result:**

1. **Distribution**:
   * The synthetic data (orange crosses) and real data (blue dots) express a similar distribution within the first two principal components.
   * The overlap between synthetic and real data points displays that the synthetic data captures the general structure of the real data.
2. **Separation**:
   * There are some points where real and synthetic data split, particularly on the outer edges.
   * This separation assists maintain privacy by verifying that synthetic data is not an exact replica of the actual data.

**t-SNE (t-Distributed Stochastic Neighbor Embedding) Result:**

1. **Cluster Structure**:
   * Both real and synthetic data structure similar clusters, suggesting that the synthetic data maintains the local form of the real data.
   * The tight overlap of synthetic and real data points within clusters specifies that synthetic data maintains the inherent patterns present in the real data.
2. **Outliers**:
   * There are a small number of outliers where real and synthetic data points do not overlap.
   * This divergence contributes to privacy, as it prevents easy identification of individual real data points.

**Summary:**

* **Alignment**:
  + Both PCA and t-SNE visualizations reveal a fine alignment between synthetic and real data distributions. The synthetic data captures the overall structure and clustering patterns of the real data.
* **Privacy Preservation**:
  + The minor deviations and outliers in the synthetic data enhance privacy by ensuring it is not a direct replica of the real data.
  + The synthetic data's ability to sustain general trends while introducing some variations is beneficial for privacy.
* **Utility**:
  + The overlap in cluster structures shows that the synthetic data preserves significant analytical value and utility. It can be applied for related analyses as the real data without sacrificing privacy.

**Conclusion:**

The synthetic data signifies a satisfactory balance between preserving the general structure and patterns of the real data while introducing enough variation to preserve privacy. The visualizations certify that the synthetic data is suitable for analytical goals, ensuring that it is not simply identifiable and fortifies the privacy of the original data.

### Anomaly Detection Results

A graph with lines and numbers

Description automatically generated with medium confidence

Figure 12:Real VS synthetic data with anomalous data excluded

**Deviation Alignment:**

1. **Initial Phase (Steps 0 to 2)**:
   * **Real Data**: Notable rise, peaking around step 1.
   * **Synthetic Data**: Steep rise, peaking slightly earlier and stabilizing.
   * **Alignment**: Both show an uphill trend initially, with synthetic data peaking earlier and stabilizing, which is tolerable for privacy.
2. **Middle Phase (Steps 2 to 12)**:
   * **Real Data**: Fluctuates with peaks and troughs.
   * **Synthetic Data**: Moderately reduces and stabilizes with minor fluctuations.
   * **Alignment**: The synthetic data's stability and absence of fluctuations help in preventing simple identification, serving privacy grounds.
3. **Final Phase (Steps 12 to 16)**:
   * **Real Data**: Sudden drop towards the end.
   * **Synthetic Data**: Stays flat.
   * **Alignment**: The synthetic data's lack of a sharp drop aids in maintaining privacy by not mimicking the exact trend.

**Deviation Analysis:**

1. **Initial Deviation**:
   * Synthetic data peaks earlier and stabilizes quicker than real data.
   * **Impact**: Good for privacy as it avoids exact replication of real data's early trends.
2. **Middle Deviation**:
   * Synthetic data lacks real data's variability.
   * **Impact**: Reduces risk of identification by not following the same fluctuations, enhancing privacy.
3. **Final Deviation**:
   * Synthetic data does not reflect the sudden drop in real data.
   * **Impact**: Helps in privacy preservation by not imitating extreme changes in real data.

**Recommendations:**

* **Trend Capture with Privacy**: Maintain a balance between trend capture and privacy. Minor deviations are useful for privacy.
* **Controlled Variability**: Introduce controlled variability to synthetic data to make it realistic enough for analysis but different enough to protect privacy.
* **Avoid Extreme Changes**: Prevent synthetic data from mimicking extreme changes in real data to further enhance privacy.

**Conclusion:**

The deviations between the "Synthetic Improved" data and the real data are beneficial for privacy purposes. While synthetic data follows general trends, it avoids exact replication of the real data's fluctuations and sudden changes, lowering the risk of identification. This steadiness confirms the synthetic data is useful for analysis while preserving the privacy of the original data.

**Strengths:**

* **Trend Capture**: The "Synthetic Improved" data captures the starting rising trend of the "Real" data, showing the model's ability to learn and mimic this aspect.
* **Stabilization**: The synthetic data stabilizes quickly after the initial rise, indicating a more manageable output from the model. This can be favorable in applications where stability is preferred.

**Weaknesses:**

* **Initial Deviation**: The initial sharp rise and subsequent stabilization of the synthetic data suggest it does not fully capture the variability seen in the "Real" data.
* **Lack of Variability**: The synthetic data seems much smoother and less variable than the real data, which continues to show fluctuations throughout the period. This indicates that the model may not be capturing all the nuances and inherent volatility in the real data.
* **Final Phase Divergence**: The significant drop in the "Real" data towards the end is not reflected in the synthetic data, suggesting that the model may encounter with capturing or predicting sudden changes or extreme values.

Also, here you can find another rolling window showcasing different behavior, as previously mentioned in the synthetic data including anomalies:

A graph showing a line graph

Description automatically generated with medium confidence

### Evaluation of Model Performance on Classification and Regression Tasks

# A graph of a line and a line Description automatically generated with medium confidence

Figure 13: Assessing Fidelity using accuracy and ROC area under the curve.

On training data, accuracy rapidly increases and stabilizes around 70-75% after approximately 50 epochs. That means that the model quickly learns to discriminate between classes in the training data.

On the test data it quickly increases to approximately 60% after 50 epochs and then fluctuates but remains relatively stable. This indicates the model's good generalization performance on unseen data. On the other hand, the training accuracy and AUC stabilize a little later, indicating the model has likely converged and is no longer learning significantly from the training data.

A graph of a graph of a graph

Description automatically generated with medium confidence

Figure 14: Assessing Usefulness using train and test on real and synthetic data and vice versa.

**The model trained on synthetic data and tested on real data shows higher test errors compared to the model trained and tested on real data. This suggests that the synthetic data does not perfectly capture the characteristics of the real data and it is our most important challenge to solve in the future.**

The model though, shows improvement in both training and test real data. The decreasing MAE on both sets indicates better generalization and usefulness of real data for training.

# Future Work and Potential Improvements

It is very important to improve model performance, avoiding potential overfitting. At the same time the biggest challenge is for our generated data to be more representative of the real data. Our future work is to focus more on the time series characteristics to understand potential trends or seasonality and adjust our model accordingly. Our long-term goal is to use the anomalies that we detected and dropped from our dataset, to create synthetic data on them based on their distribution and amend them back to our synthetic data pool.

# References

* Confident AI. (2023). How to Generate Synthetic Data Using LLMs - Part 1. Retrieved from <https://www.confident-ai.com/blog/how-to-generate-synthetic-data-using-llms-part-1>
* Financial Conduct Authority. (2020). Using Synthetic Data in Financial Services. Retrieved from <https://www.fca.org.uk/publication/corporate/report-using-synthetic-data-in-financial-services.pdf>
* Forbes. (2024). Scaling ML with LLMs: From Data Labeling to Synthetic Dataset Creation. Retrieved from <https://www.forbes.com/sites/forbestechcouncil/2024/03/13/scaling-ml-with-llms-from-data-labeling-to-synthetic-dataset-creation/>
* Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative Adversarial Nets. Advances in Neural Information Processing Systems, 27. Retrieved from <https://arxiv.org/abs/1406.2661>
* Perarnau, G. (2017, March 20). Fantastic GANs and where to find them. Guim Perarnau Blog. Retrieved July 12, 2024, from <https://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them>
* Heka AI. (2023). Generating Synthetic Time-Series Using GANs and Probabilistic Auto Regressive Neural Networks. Retrieved from <https://heka-ai.medium.com/generating-synthetic-times-series-using-gans-and-probabilistic-auto-regressive-neural-networks-a82200f2c73b>
* Jansen, S. (2022). Generative Adversarial Nets for Synthetic Time Series Data. Retrieved from <https://stefan-jansen.github.io/machine-learning-for-trading/21_gans_for_synthetic_time_series/>
* JPMorgan. (2023). Using GANs for Sharing Networked Time Series Data: Challenges, Initial Promise, and Open Questions. Retrieved from <https://www.jpmorgan.com/technology/artificial-intelligence/initiatives/synthetic-data>
* Medium - Microsoft. (2023a). Synthetic Data Generation Using Generative Adversarial Networks (GANs) - Part 1. Retrieved from <https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-1-47ecbf46b575>
* Medium - Microsoft. (2023b). Synthetic Data Generation Using Generative Adversarial Networks (GANs) - Part 2. Retrieved from <https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-2-9a078741d3ce>
* Towards AI. (2023). GANs for Synthetic Data Generation. Retrieved from <https://towardsai.net/p/l/gans-for-synthetic-data-generation>
* Towards Data Science. (2023). How to Generate Synthetic Tabular Data. Retrieved from <https://towardsdatascience.com/how-to-generate-synthetic-tabular-data-bcde7c28038a>
* Yoon, J., Jarrett, D., & van der Schaar, M. (2019). Time-series Generative Adversarial Networks. Advances in Neural Information Processing Systems, 32. Retrieved from <https://proceedings.neurips.cc/paper_files/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf>
* Sharifi, N. (n.d.). Generative AI: Synthetic data generation with GANs using PyTorch. Towards Data Science. Retrieved July 12, 2024, from <https://towardsdatascience.com/generative-ai-synthetic-data-generation-with-gans-using-pytorch-2e4dde8a17dd>
* XQ (n.d.). Exploring GANs to generate synthetic data. Medium. Retrieved July 12, 2024, from <https://medium.com/the-research-nest/exploring-gans-to-generate-synthetic-data-ca48f8a4b518>
* Tan, L. (2021, February 19). Generating synthetic tabular data. Towards Data Science. Retrieved July 12, 2024, from <https://towardsdatascience.com/generating-synthetic-tabular-data-503fe823f377>
* Rochetti, M., De Gregorio, A., Basile, C., & Osmani, V. (2024). Protect and extend - Using GANs for synthetic data generation of time-series medical records. arXiv. <https://arxiv.org/pdf/2402.14042>
* Mohammadi Mahmood (2021, Jun 15). Synthetic data generation using Generative Adversarial Networks (GANs) Part 2. Medium. Retrieved July 12, 2024, from <https://medium.com/data-science-at-microsoft/synthetic-data-generation-using-generative-adversarial-networks-gans-part-2-9a078741d3ce>
* Ali, P. J. M., Faraj, R. H., Koya, E., Ali, P. J. M., & Faraj, R. H. (2014). Data normalization and standardization: a technical report. Mach Learn Tech Rep, 1(1), 1-6.
* Nikitin, A. (2023, October 19). Time series augmentations. Towards Data Science. Retrieved July 12, 2024, from <https://towardsdatascience.com/time-series-augmentations-16237134b29b>
* Xu, L., Skoularidou, M., Cuesta-Infante, A., & Veeramachaneni, K. (2019). Modeling tabular data using Conditional GAN. In Advances in Neural Information Processing Systems (NeurIPS 2019). Retrieved from <https://proceedings.neurips.cc/paper_files/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf>
* Jiang, H., Kim, B., Guan, T., & Le, Z. (2018). Clustering and unsupervised anomaly detection with l2 normalized deep auto-encoder representations. In 2018 IEEE International Conference on Data Mining (ICDM) (pp. 1002-1007). IEEE. <https://ieeexplore.ieee.org/document/8489068>
* Xu, C., Skoularidou, M., Cuesta-Infante, A., & Veeramachaneni, K. (2022). GT-GAN: General Purpose Time Series Synthesis with Generative Adversarial Networks. In Advances in Neural Information Processing Systems (NeurIPS 2022). Retrieved from <https://proceedings.neurips.cc/paper_files/paper/2022/hash/f03ce573aa8bce26f77b76f1cb9ee979-Abstract-Conference.html>
* Yoon, J., Jarrett, D., & van der Schaar, M. (2024). TimeGAN: Time-series Generative Adversarial Networks. In 2024 IEEE/IFIP Network Operations and Management Symposium (NOMS). Retrieved from <https://leris.dcomp.ufscar.br/wp-content/uploads/2024/01/Noms_2024___TimeGAN.pdf>
* Nikitin, A. (2023, October 19). Modeling and generating time series data using TimeGAN. Towards Data Science. Retrieved July 12, 2024, from <https://towardsdatascience.com/modeling-and-generating-time-series-data-using-timegan-29c00804f54d>
* Brophy, E., Wang, Z., She, Q., & Ward, T. (2023). Generative Adversarial Networks in Time Series: A Systematic Literature Review. ACM Computing Surveys, 55(10), 1-31. <https://doi.org/10.1145/3559540>
* Ashrafi, N., Schmitt, V., Spang, R. P., Möller, S., & Voigt-Antons, J.-N. (2024). Protect and extend - Using GANs for synthetic data generation of time-series medical records. arXiv. <https://arxiv.org/html/2402.14042v1>
* Tan, J., Yang, J., Wu, S., Chen, G., & Zhao, J. (2021). A critical look at the current train/test split in machine learning. arXiv. <https://arxiv.org/abs/2106.04525>
* Lampis, A., Lomurno, E., & Matteucci, M. (2023). Bridging the Gap: Enhancing the Utility of Synthetic Data via Post-Processing Techniques. arXiv. <https://arxiv.org/abs/2305.10118>

# Appendices

## Appendix 1

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Adversarial Training Phase** | **Joint Training Phase** |
| **Objective** | Focus on adversarial loss between generator and discriminator | Optimize adversarial, reconstruction & embedding losses |
| **Training Process** | Alternates between the update of the generator and discriminator | Simultaneously update of the model towards balancing multiple objectives |
| **Complexity** | Single focus 🡪Relatively straightforward | More complex 🡪 Involves balancing multiple losses |
| **Primary Goal** | Improvement of the generator's ability to fool the discriminator | Ensures the realism and the preservation of temporal dynamics of the generated data |
| **Loss Functions** | Adversarial loss only | Adversarial loss, reconstruction loss & embedding loss |
| **Temporal Dependency** | Focuses on realism 🡪 May not fully capture temporal dependencies | Ensures that temporal relationships and dependencies are maintained |
| **Generator Performance** | Trained 🡪 Producing realistic time-series data | Trained 🡪 Producing realistic and temporally accurate data |
| **Discriminator Role** | Distinguishes 🡪 Between real and generated data | Same role as adversarial phase & evaluated within joint objectives |
| **Use Case** | Initial phase 🡪 Improve basic realism of generated data | Refinement phase 🡪 Ensuring data realism and temporal fidelity |
| **Output Quality** | High visual quality 🡪 May lack temporal coherence | High visual & temporal quality |

## Appendix 2 – Previous results (Before optimization)

A screen shot of a graph

Description automatically generated

A graph on a white background

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a graph

Description automatically generated