**Differentially Private Time Series Data Generation using**

**GS-WGAN**

**KUMARESH S**  (19Z327)

**AJITH NARAYANA S** (19Z331)

**MRIDULA M** (19Z332)

**SRAVYA VANKADARA** (19Z348)

**CHANDRA PRAKASH J** (20Z461)

**19Z820 PROJECT WORK–2**

Dissertation submitted in partial fulfilment of the requirements for the degree of

**BACHELOR OF ENGINEERING**

**Branch: COMPUTER SCIENCE AND ENGINEERING**

of Anna University



**APRIL 2023**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**PSG COLLEGE OF TECHNOLOGY**

(Autonomous Institution)

**COIMBATORE – 641 004**

**PSG COLLEGE OF TECHNOLOGY**

(Autonomous Institution)

**COIMBATORE - 641 004**

**SYNTHETIC TIME SERIES DATA GENERATION USING**

**GS-WGAN**

Bona fide record of work done by

**KUMARESH S** (19Z327)

**AJITH NARAYANA S** (19Z331)

**MRIDULA M** (19Z332)

**SRAVYA VANKADARA** (19Z348)

**CHANDRA PRAKASH J** (20Z461)

Dissertation submitted in partial fulfilment of the requirements for the degree of

**BACHELOR OF ENGINEERING**

**Branch: COMPUTER SCIENCE AND ENGINEERING**

of Anna University

**APRIL 2023**

**…..……………………… ....……..………………………**

**Dr. K. Sathiyapriya Dr. G. Sudha Sadasivam**

Faculty guide Head of the Department

Certified that the candidate was examined in the viva-voce examination held on …………….

……………………. ..…………………….

(Internal Examiner) (External Examiner)

**CERTIFICATE**

Certified that this report titled **‘Synthetic Time Series Data Generation Using GS-WGAN’** for the Project Work 2(19Z820) is a bonafide work of **Kumaresh S (19Z327), Ajith Narayana M S (19Z331), Mridula M (19Z332), Sravya Vankadara (19Z348),** and **Chandraprakash J (20Z461)** who have carried out the work under my supervision for the partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering. Certified further that to the best of my knowledge and belief, the work reported herein does not form a part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion.

|  |  |
| --- | --- |
| **Place : Coimbatore**  **Date :** | **Dr. K. Sathiyapriya**  **Assistant Professor**  **(Selection Grade)**  **Department of Computer Science**  **PSG College of Technology** |

COUNTERSIGNED

**HEAD**

**Department of Computer Science and Engineering**

**PSG College of Technology**

**COIMBATORE – 641 004**

**SYNOPSIS**

Privacy has become a major concern in the past few years with the rise of machine learning and deep learning models being vulnerable to invasion attacks. The humongous amount of data being used to train these models is susceptible to leakage. This is due to the model’s capability of retaining training information. Much research effort was dedicated to exploring differential privacy (DP) as a means of privacy preservation.

DP is a mathematically robust guarantee. It ensures that the outcome of a differentially private analysis will draw the same conclusion about any individual's private information, regardless of whether that individual's private information is included. To generate DP synthetic data, Generative Adversarial Network (GAN) models are widely utilized. This gave rise to a plethora of GAN variants.

This project focuses on generating differentially private time series data by utilizing the capabilities of the GAN models. With tabular data generation using DP GANs being a much-underexplored topic, we propose to generate tabular time series data using a modified version of DP Wasserstein GAN (DP-WGAN). The results are compared with TimeGAN and (Continuous- Recurrent Neural Network GAN) C-RNN-GAN based on privacy and utility trade-offs. The synthetic dataset’s performance and quality are tested. This project aims to use the Wasserstein loss function-based GAN to produce differential private synthetic time series data.

**TABLE OF CONTENTS**

**Chapter Page No.**

**SYNOPSISi**

**1. INTRODUCTION1**

1.1 Synthetic Data Generation1

1.2 Differential Privacy2

1.3 Wasserstein Distance2

1.4 Differentially Private-Stochastic Gradient Descent2

1.5 Deep Learning Models3

1.6 Motivation5

1.7 Problem Statement6

1.8 Objective6

1.9 Scope6

**2. LITERATURE STUDY7**

**3. SYSTEM ANALYSIS9**

3.1 Hardware Requirements9

3.2 Software Requirements9

3.3 Dataset 10

3.4 Functional Requirements 10

3.5 Non-Functional Requirements10

3.6 Feasibility11

**4. SYSTEM DESIGN AND IMPLEMENTATION9**

**BIBLIOGRAPHY12**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LIST OF FIGURES** | | | | |
|  | | | | |
| **Figure No.** | | **Figure Name** | **Page No.** | |
|  |  | | |  |
| 1.1 | GAN Architecture | | | 3 |
| 1.2 | GS-WGAN Architecture | | | 5 |
| 1.3 | Fed-GS-WGAN Architecture | | | 5 |

**CHAPTER 1**

# **INTRODUCTION**

Machine Learning and Deep Learning models are now on the rise, with many enthusiasts utilizing these technologies daily. Many businesses are adopting these algorithms into their process, adapting their application to meet their needs [1]. With the level of interest in these ever-evolving methodologies, more resources in terms of processing and data are necessary to produce better results. Computational power has increased dramatically in recent years. New Central Processing Units (CPUs) and Graphics Processing Units (GPUs) emerging every year show much higher performance than in prior years. Data, on the other hand, has always been plentiful. The inability to securely manipulate and represent this immense volume of information is a fundamental reason for the domain's inhibition [2].

Every company and person prioritize security. Although most firms utilize relative model architectures for their use cases, sharing these trained models publicly is never done, even though it might lead to better-trained models. This is owing to the possibility of security attacks on the models, which might result in the disclosure of data used to train these models [3][4]. Model inversion attacks can recover the model's training data from the model parameters [5]. Because many firms would use sensitive data to train the model, exposing such data would result in legal disputes and a loss of confidence. An example would be hospitals using a patient's data for training a machine learning model.

Currently, security measures against model exploits are offered by either redesigning model architectures to a more secure model [6] or training models with synthetic data [7]. The former corresponds to a considerably more time-consuming operation. This is because transitioning from an existing architecture to a newer one would be challenging. On the other hand, the latter might result in less model utility.

**1.1 Synthetic Data Generation (SDG)**

Synthetic data is created and annotated artificial data that closely resembles real-world data. SDG is typically used to fill gaps in datasets or to replace highly sensitive data. Corporations utilize synthetic data to train their models to ensure privacy and prevent data leaks. Although synthetic data closely resembles the features of the real-world data to be substituted, they are not identical owing to the privacy-utility trade-off [8]. The privacy-utility trade-off logically explains that as the synthetic and real data became increasingly comparable, the privacy measure of the synthetic data would fall but the model's utility would increase. Similarly, increasing the privacy measure of the synthetic data causes the similarities between the produced and actual data to become sparse, resulting in poorer model usefulness. The best point in the privacy-utility trade-off must be identified based on the needs of the company.

* 1. Differential Privacy

Differential Privacy (DP) is the cornerstone of many algorithms that apply privacy safeguards [9]. DP is a mathematical definition of privacy that consists of statistical and machine learning constraints. DP divides information into two categories: general information and private information. According to established DP perspectives, general information is information inferred from the entire population of the dataset. Contrarily, private information is the difference in data before and after removing an individual's data from the dataset. This definition of private information resulted in the term differential.

DP mathematically assures that conclusions regarding an individual's private information will be the same whether or not the individual's private information is included in the differentially private analysis. A randomized mechanism with range R is (,)-DP, if

|  |  |
| --- | --- |
|  | *(1)* |

Holds for any subset and for adjacent datasets and , where both differs by one training sample as shown in formula (1). is the GAN training algorithm and corresponds to the upper bound of privacy loss whereas corresponds to the probability of breaching DP constraints.

DP can only guarantee the privacy of an individual’s private information and not general information which can be inferred from the entire population. DP includes useful qualities such as the bounding of privacy loss, group privacy, the composition of DP attributes, and post-processing closure.

Post Processing is a pivotal property of DP. It ensures that without additional knowledge about the private database, one cannot compute a function of the output of a private algorithm. Post processing also warrants that the attackers cannot make the algorithm less differentially private.

* 1. Wasserstein Distance

Wasserstein Distance (WD) is comparable to a cost function in that it is determined using the least amount of work [10]. It is the amount of work required to transfer a segment of the graph from one distribution to another. The amount of work required to shift represents the amount of work required to find similarities between distributions. As a result, more work would be necessary for highly different distributions indicated by a higher WD. Because WD is a distance measure, it may be used for a broad variety of Machine Learning problems that can be described in metric space. The key benefit of WD over other distance approaches is that it can be used with any kind of data and distribution. The only criteria to using WD is to discretely representing the distribution in a metric space.

* 1. Differentially Private-Stochastic Gradient Descent (DP-SGD)

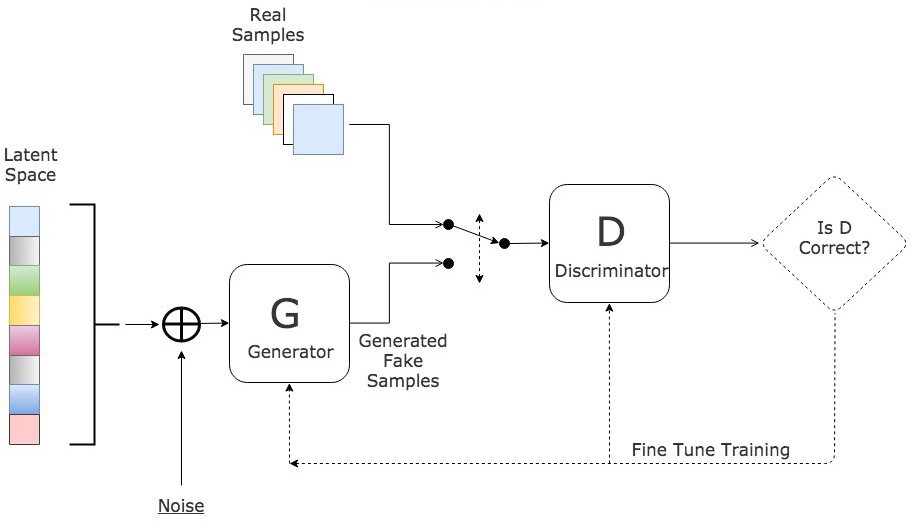
The widely utilized weight updation method of stochastic gradient descent and the characteristics of DP are combined in Differentially Private-Stochastic Gradient Descent (DP-SGD) [11]. By including noise during the model's training phase, this strategy alters the SGD process. The privacy measure provided by the model depends on how much noise is added to the parameter during weight optimization. The least amount of noise necessary to cover the biggest gradient while still ensuring the privacy of each sample in the batch would be the ideal amount of noise to add. Gradient clipping limits the gradients depending on the clipping threshold. It is determined by user-defined functions and is a method used to account for the existence of outliers. As a result, the model would have to calculate the parameter gradients for each sample in a batch. This results in the per-sample gradient. Similarly, weight clipping is used in DP-WGAN. The micro-batch approach is utilized, where the batch size is considered as one, to save computation time.

* 1. Deep Learning Model

This section describes briefly some of the deep learning models implemented in the system.

* + 1. Generative Adversarial Networks (GANs)

GANs are generative models that are used for unsupervised learning tasks in machine learning applications. The goal of such a model is to create new data from an original input dataset. It is used to generate datasets replicating patterns or regularities that are present in the original dataset. The GAN is made to learn and discover these patterns by using an innovative way of breaking the unsupervised learning problem into a supervised learning problem. This is done by using two sub-models named the generator and the discriminator. The general functioning of how these sub-models’ function can be depicted using the Fig 1.1.



**Figure 1.1 GAN Architecture**

The generator model is trained to generate new data while the discriminator model is made to classify whether it is fake or not. The generator trains to make the discriminator believe the data is real. The discriminator contrarily trains to classify, whether the data is original or generated, more accurately.

To be more precise, the generator model creates a sample in the domain using a fixed-length random vector as input. The generative process is seeded with a random vector that is taken from a Gaussian distribution. A compressed representation of the data distribution will be formed. This is done after training when points in this multidimensional vector space match points in the problem domain.

The loss of the generator function, as shown in formula (2), is calculated in order to train the model. It uses a Binary Cross Entropy loss function.

|  |  |
| --- | --- |
|  | *(2)* |

Where L(G) stands for generator loss, stands for the ability of the discriminator to find differences between real and fake images, and stands for the ability of the generator to generate real images. Both the generator and the discriminator use the same loss function. The loss function is based on the discriminator’s ability to figure out whether the image is real or synthetic.

The discriminator is a typical classification model. It predicts a binary class label of real or false based on an example from the domain. The training dataset contains the real examples combined with the generator model’s fake examples.

In this way the two models compete with each other until the generator becomes good at creating synthetic data and can fool the discriminator.

* + 1. Wasserstein Generative Adversarial Networks

Wasserstein GAN (WGAN) was designed as an improvement over the existing GAN architecture. It provides better stability during the training phase of the model and represents the quality of the generated images through its loss function [12]. WGAN is closely related to the well-established standard deep convolutional GAN (DCGAN). The differentiating factor of WGAN was the replacement of the discriminator with the critic. Rather than classifying images as real or fake based on probabilities, the critic scores the images based on the realness and fakeness of an image. Training the generator must lead to the minimization of the distance between the distribution of the generated samples and training data. WGAN uses the Wasserstein distance to find the difference between the actual and generated distributions. The WGAN uses Gradient Clipping for the critic model and updates the critic model more frequently than the generator. Gradient Clipping led to much longer training time in WGAN. Therefore, the gradient penalty method is currently used by the standard WGAN, ensuring smooth training.

* + 1. Gradient Sanitized-Wasserstein Generative Adversarial Network (GS-WGAN)

Gradient Sanitization is a technique which is used to ensure DP training to a generator. GS is performed by clipping the gradient and adding noise to the gradient [13] as shown in formula (3).

|  |  |
| --- | --- |
|  | *(3)* |

Where stands for technique, gt stands for the gradient to be clipped and C stands for the clipping value.

Since the clipping value of the gradient is extremely sensitive to the hyperparameters used, it is a challenging task to find the optimal value. An exhaustive search is required to find the best clipping value for a given set of hyperparameters. GS-WGAN performs selective sanitization which pertains only to a set of parameters. This led to a more reliable training of the discriminator. To further bound the sensitivity of the model’s optimizer, the Wasserstein metric is used as a loss function to calculate the clipping value. Lower variance in gradient norms was achieved during training due to the WD loss function. Further, privacy amplification was performed by implementing subsampling of the database. Including Federated Learning, two different architectures of GS-WGAN was implemented using the ResNet model as shown in Fig 1.2 and Fig 1.3.

|  |  |
| --- | --- |
|  |  |
| Figure 1.2 GS-WGAN | Figure 1.3 Fed-GS-WGAN |
|  |  |

* 1. Motivation

Extreme data requirements have arisen as a result of the recent development in data-intensive algorithms. Although improved performance is promoted, a significant quantity of data and processing power is needed. Due to the enormous volume of data being used, the number of vicious attackers looking to obtain and exploit this data has skyrocketed. Recent data breaches like the Log4Shell vulnerability have caused the technology industry to turn its attention to more secure methods. Although researchers are always trying to add security features to present algorithms, synthetic data generation has been encouraged similarly to ensure privacy. This is because SDG and federated learning algorithms can utilize current technologies. This minimizes the laborious process of migrating to a different system. Additionally, by regulating the privacy-utility trade-off, privacy via data production allows the user better control over limiting the privacy loss of data. Due to less demanding training needs, GANs have emerged as excellent models when it comes to DP SDG. In comparison to VAEs and other deep-learning models, the recently published GS-WGAN has shown much potential for producing DP synthetic image data.

* 1. **Problem Statement**

Rapid growth of Deep Neural Networks (DNNs) has given rise to interest in the usage of data collected by large institutions. While such data sources can prove to significantly aid the progress of DNNs, they are often restricted by privacy and confidentiality concerns. Dataset synthesis is widely viewed as a solution for the restrictions in accessing large scale data. Many different approaches have been taken to implement SDG with privacy guarantees, most prominent one being GAN. Application of mathematical frameworks of privacy make GANs privacy. However, most attention and implementation of GANs has been in image/image adjacent datasets. It is imperative to also consider other prominently used dataset types, particularly, tabular data. Furthermore, synthesis of time series data through GAN models using multiple critics has also not garnered much attention. If established privacy-preserving GAN models capable of generating image datasets could be generalised to also accept and output tabular, time series data, then the scope of SDG would grow by leaps and bounds.

* 1. Objective

The main objectives of the project are to:

* Generate synthetic tabular time series data utilizing conventional Deep Learning models
* Utilize GS-WGAN architecture for providing differential privacy to training data
* Observe and record utility-privacy tradeoffs through downstream model utility and data similarities
* Compare proposed GS-WGAN model with other types of GAN for tabular time series data
  1. Scope

The GS-WGAN architecture can be extended to many other types of data, making it compatible to accept different forms of data. Adversarial methodologies could lead to prolonged training time requirements that can be potentially handled by better gradient clipping methodologies. Adding noise to the SDG though leads to privacy causes the model utility to decrease and increase training time. Furthermore, handling biased and highly erroneous data is a challenging task that can be solved by focusing more on reproducing original data characteristics. Identifying newer avenues to add noise, such as colour channels of images to provide privacy, could be further delved into.

**CHAPTER 2**

# **LITERATURE STUDY**

Synthetic data generation (SDG) has given hope to many to be the solution for large scale data requirements [7]. In particular, the increasing usage of deep learning algorithms in various domains for both broad applications as well as hyper-specific use cases has led to higher visibility on the regulation, distribution, and publication of large-scale data [1] [26]. Many scholars, researchers, and domain-experts have performed extensive and exhaustive research on the kinds of data required to diversify the potential held by synthetic data generation. Currently, the landscape of SDG methods is constantly changing, from auto-encoders, GANs to transformers. Due to the comparatively less tedious computational effort needed to train, test, and publish GAN models, they are preferred in environments requiring dataset generation.

Deep learning models are susceptible to different types of attacks that threaten the privacy and utility of their outcomes [5][6]. Ximeng Liu et.al [3] outlines many of the threats and attacks posed to Deep Neural Network models (DNNs). Hui Sun et.al [4] describe the adversarial attacks that can be used against GANs, providing a clearer understanding of model inversion attacks that attempt to recover confidential information from the model outputs. In recent years, differential privacy has become a trustable mathematical framework that can be relied upon to extend the privacy guarantees of DNNs, as well as maintain utility in the data generated [9]. Aiming to implement differential privacy measures that protect against exposure of private, sensitive information, Martin Abadi et.al [14] proposed a Differentially Private Stochastic Gradient Descent (DP-SGD) method for training DNNs with a modest total privacy loss. Alternatively, Nicolas Papernot et.al [15] proposed the Private Aggregation of Teacher Ensembles (PATE) approach for training models using disjoint datasets whose results also exhibit differential privacy properties.

With multiple ways of training and testing DNNs in a differentially private manner, GANs also started to develop by adopting these guarantees. Due to their varied applications and usage scenarios, many types of differentially private GANs have emerged to meet the demands in the industry. Liyue Fan [16] surveys various GAN variants, including Conditional GAN (CGAN) [17], Wasserstein GAN (WGAN) [12][18], DPGAN [19], dp-GAN [20], GANobfuscator [21], PATE-GAN [22], SPRINT-GAN [23].etc. Each of these variants can be distinguished through their training method, differential privacy approach, or application domain.

In their survey of over 100 published papers, Abraham Wu et al. [24] identify 26 different applications for GANs in different architectural and constructional contexts. Further, they posit that the primary limitation to advancing research on GANs and diversifying their usage is the lack of high quality datasets specific to application environments. Alvaro Figueira et al. [25] discuss synthetic data generation using GANs for tabular data. They conclude that major breakthroughs in synthetic data generation using GANs has been in image generation, while tabular data generation is a field largely unexplored in comparison.

Wasserstein GAN, proposed by Martin Arjovsky et.al [18], improves learning stability of GANs by using the Wasserstein distance as a cost function [10]. WGAN improved the feasibility provided for debugging and hyperparameter search. However, these advances did not guarantee to be differentially private. Dingfan Chen et.al [13] provided these guarantees by distorting gradient information in a precise manner, putting forth the Gradient Sanitized Wasserstein GAN (GS-WGAN) model with rigorous privacy guarantees that could be trained in both centralized and federated environments. Validated on benchmark image datasets, GS-WGAN applies the privacy preserving training techniques to the generator alone, such that sanitized datasets can be generated while maintaining downstream utility.

Liyang Xie et al. [29] proposed a framework implementing differential privacy for WGAN, using noise addition and weight clipping. The authors demonstrate the effectiveness of their approach on several datasets, including the MNIST and MIMIM-III datasets, and show that their differentially private GAN is able to generate high-quality synthetic data while preserving the privacy of individuals in the training data. Jinsung Yoon et al. [30] introduce the concept of supervised loss through TimeGAN, which is different from other GAN architectures (e.g. WGAN) where unsupervised adversarial loss is implemented on both real and synthetic data. The TimeGAN model is encouraged to capture time conditional distribution within the data by using the original data as a supervision. Additionally it contains an embedding network, which is in charge of decreasing the dimensionality of the adversarial learning space.

Although GANs have gained popularity for their proficiency in image generation tasks, many datasets and data requirements seek the usage of tabular data. This type of data tends to often be multi-modal in nature, and some of the prominent usages of synthetic tabular data lie in time-series applications. Pepijn te Marvelde [27] et.al harnessed the privacy guarantees of GS-WGAN to generate synthetic time-series data, concluding that the convolutional architecture for GS-WGAN allowed employing of image based GANs for time series data synthesis. Valtteri Nieminen [28] used GS-WGAN for the synthesis of tabular data, and provided privacy guarantees by using the subsampling technique during the training process. Diversification of the data that can be accepted and generated by established GAN models that are proven to produce data with privacy preserving measures and adequate downstream utility would pave the way to solving much more complex data requirement problems and many more advancements in the field of deep learning.

The study served the purpose of understanding the various approaches to synthetic data generation, usage of GANs, privacy preservation in GANs, and tabular data synthesis using GANs. From this, it is observed that differential privacy guarantees have been widely explored in synthetic data generation, particularly for image-based datasets. Due to the wide ongoing research endeavours in GAN applications, GAN models published could tend to be inflexible and generalizable in the context of kinds of data accepted. WGAN trained in a Gradient Sanitized method (GS-WGAN), has proven to be effective in generating image-based data after training on benchmark datasets. It is also capable of generating time-series image data while maintaining downstream utility and privacy guarantees. Variations to the GS-WGAN to accept and generate tabular time series data can be analysed, and its utility-privacy trade-off can be measured through the synthesised dataset’s application. [8]

**CHAPTER 3**

# **SYSTEM ANALYSIS**

This chapter focuses on the hardware and software requirements essential to develop, train, test, implement the system and its modules. It also discusses the datasets used, along with the feasibility of the system.

1. **Hardware Requirements**

The hardware requirements mentioned are the minimum hardware settings required to deploy the project in a computer system.

1. **OS:** Windows, Mac, or Ubuntu
2. **Minimum Cores:** 6
3. **Graphical Processor:** 2 GB RAM
4. **Minimum System RAM:** 16 GB
5. **Hard Disk Space:** 30 GB SSD
6. **GPU:** NVIDIA 2070
7. **Software Requirements**

The software requirements mentioned are the minimum software setting necessary to run the project in a computer system.

1. Python 4.8 or above
2. Google Colaboratory
3. Python Libraries:
   1. **NumPy:** NumPy is a library for the Python language. It is a general-purpose array processing package that provides support and high-level mathematical operations for working on large, multi-dimensional arrays and matrices.
   2. **pandas v.1.4.2+:** pandas is a Python library that is used to analyze and manipulate data. It provides fast and flexible data structures to ease working with tabular data.
   3. **PyTorch v.1.7.0+:** PyTorch is a Python library based on the Torch library. It provides Tensor computing capabilities and deep neural networks.
   4. **TensorFlow 2.1.0:** TensorFlow is a library made by Google for easing the process of building and deploying artificial intelligence and machine learning models
   5. **Google Colaboratory:** Google Colaboratory is a cloud-run Jupyter notebook environment that allows collaboration on code without requiring additional setup. It supports man popular machine learning libraries and even provides facilities to switch runtime environments.
   6. **Docker:** Docker allows the usage of virtualization to use container-as-a-service resources. It allows for easier application development as well as infrastructure management.
4. **Datasets**

The dataset contains stock prices that comprises information on all the variation in the ADANI GREEN stock price per minute. This is a Gujarat-based company that is a part of the Adani group. The information in the dataset has been collected for the days starting from 18th June, 2018 till 22nd October, 2022. The dataset consists of time-series data and is multivariate and sequential in nature. There are 6 fields and 370574 data points which are either timestamps or real numbers. The attributes taken into consideration are:

* Date: It consist of the date as well as the timestamp of the data recorded. The data is captured at a time interval of 1-minute.
* Close: It signifies the close price of the ADANI GREEN stock at the end of each minute.
* High: It depicts the highest price the stock had achieved during the 1-minute time interval. It could be greater than both the Close and Open price of that time instance.
* Low: It depicts the lowest price the stock had achieved during the 1-minute time interval. It could be lower than both the Close and Open price of that time instance
* Volume: It consists of the amount of stock shares traded during the 1-minute time interval.
* Middleband: It is used to capture the moving average of the ADANI GREEN stock. It acts as a good indicator to identify and forecast price fluctuations and volatility.

1. **Functional Requirements**

The functional requirements of the system are:

* Accept tabular dataset as input
* Generate synthetic time-series data that imparts differential privacy.
* Calculate the utility-privacy tradeoffs

1. **Non-Functional Requirements**

Non-functional requirements are requirements that specify the criteria that can be used to judge the operation of a system rather than the behavior of a system.

1. **Usability:**

The system must be able to generate synthetic data with exactly same qualities as original without any errors, while maintaining differential privacy.

1. **Efficiency:**

The system must be able to generate synthetic data more accurately, in lesser time.

1. **Correctness:**

The output of the system matches the expectations outlined in the requirements, and the system operates without failure.

1. **Feasibility:**
2. **Economic Feasibility:**

Training of modules and collection of the dataset requires hardware facilities like faster RAM, processors, GPUs, etc. The libraries and tools used for the development of the system are open-source and under a general-purpose license. The training, testing, and deploying of these modules do not require special additional peripherals.

1. **Technical Feasibility:**

The required resources and tools to develop and run the system are available on hand. They have sufficient documentation and support to perform maintenance and upgradation of the system if necessary.

1. **Operational Feasibility:**

The system satisfies operational feasibility in setups that satisfy the aforementioned hardware and software requirements. Additional or improved resources will improve the operational functioning of the system.

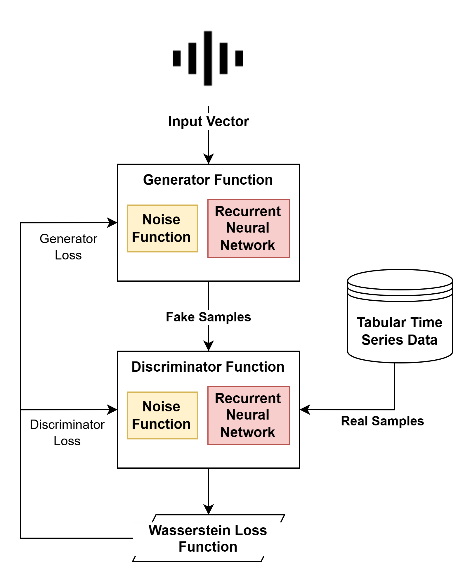
**CHAPTER 4**

# **SYSTEM DESIGN AND IMPLEMENTATION**

This chapter focuses on the hardware and software requirements essential to develop, train, test, implement the system and its modules. It also discusses the datasets used, along with the feasibility of the system.

1. **System Architecture**

The overall system architecture of the proposed Differentially Private-Time based Wasserstein GAN (DP-TGAN) is shown in figure 4.1.



**Figure 4.1 Overall architecture of DP-TWGAN**

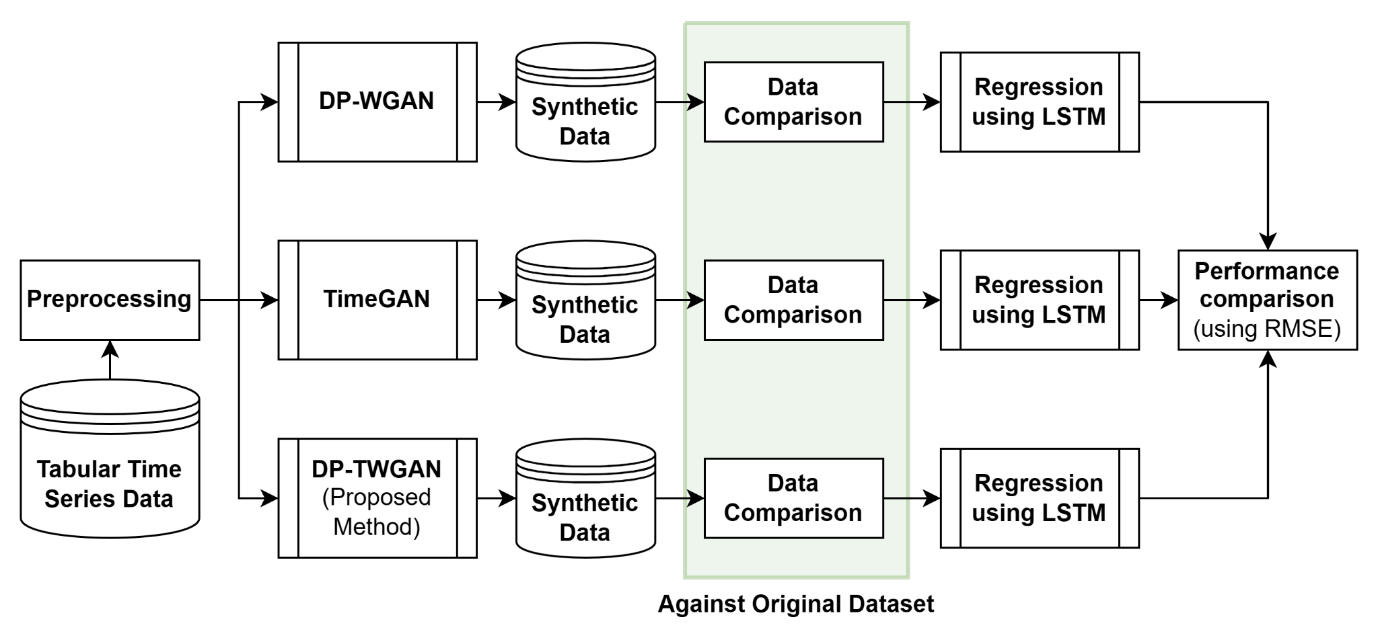
The DP-TWGAN is a modified version of its predecessor DP-WGAN. The primary modification performed over its architecture is the replacement of weight clipping with gradient sanitization and the introduction of Recurrent Neural Networks (RNNs) into the generator and discriminator modules.

Gradient Sanitization is a technique inspired from GS-WGAN which showed better results when compared with weight clipping for the convergence of differentially private Wasserstein GAN models. This is due to the fact that weight clipping has no antecedence and is a manually set value, making fine tuning very difficult. Therefore, achieving optimality and convergence of model requires large amounts of time. Gradient Sanitization on the other hand, implements gradient clipping only to a set amount of parameter, hence bounding them. Unlike its predecessor, gradient clipping converges faster due to minimum oscillation around the local minima but is computationally much intensive.

The introduction of RNNs into the model is to capture the temporal locality of dataset. RNNs are well know to hold the innate ability to serve as memory objects. Utilizing them to capture the time dependencies during the generation and discrimination of data produces better results. The disadvantage of this technique is that the privacy bound of the model decreases further since the RNN helps the model remember more data from the original dataset. Accordingly, privacy bound must be tuned to achieve an appropriate privacy-utility tradeoff.

1. Utility

The utility of the proposed model is compared with pre-existing models as shown in figure 4.2.

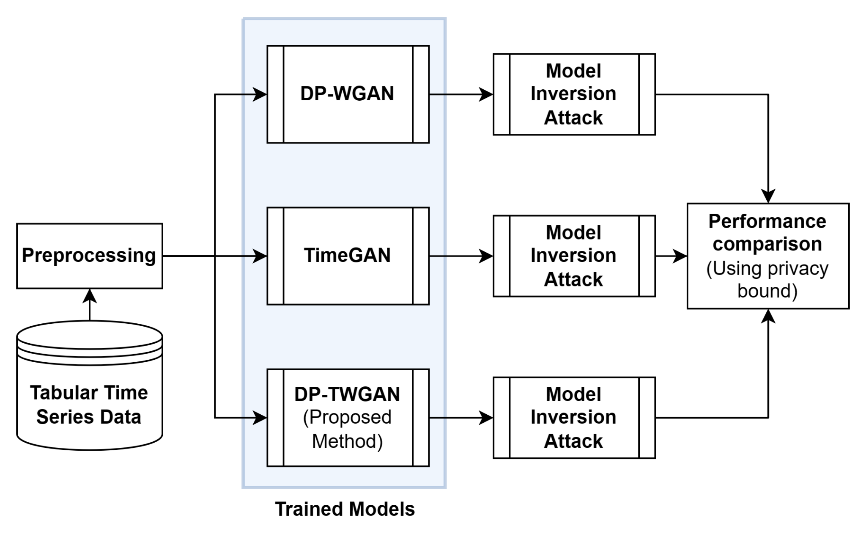


**Figure 4.2 Utility Comparison**

The Utility comparison is performed to ensure and record the privacy-utility tradeoff in a more robust manner. To measure the utility of each model, an Long Short-Term Memory (LSTM) based price prediction is performed. Each model generates synthetic data using the original ADANI GREEN dataset. After SDG, the data generated by all models are compared against the original dataset using statistical means. Through the comparison of mean, median, mode, averages, and distribution of all columns of the dataset, a definite solution is obtained in terms of statistical resemblance. After statistical comparison, the data synthesized by each model is used to train the LSTM model. Testing on the other hand is done using the original dataset to emulate real-world scenario. Based on the RMSE scores obtained for each model, the performance of the GAN models is assessed. During the performance comparison, the privacy bound of every model is set to be the same to ensure an unbiased study.

1. Privacy

The privacy of the proposed model is compared with pre-existing models as shown in figure 4.3.



**Figure 4.3 Privacy Comparison**

Privacy Comparison is performed to check the privacy bound and vulnerability of the models implemented. Though privacy scores can be calculated effectively, executing model inversion attacks on GAN models helps to better understand the weaknesses of each model in a straightforward manner.

1. **Models Implemented**
2. DP-WGAN

Differentially Private-Wasserstein GAN (DP-WGAN) is implemented based on “Object Perturbation”. “Object Perturbation” is when the objective function is introduced with noise to achieve differential privacy. In this model, the objective function is the Wasserstein GAN’s loss function. The DP-WGAN utilizes “Randomized Response” to add random noise to the discriminator’s output making it difficult for the generator to learn sensitive information about the training data. The privacy budget parameter controls the amount of noise to be added. The DP-WGAN uses weight clipping and Wasserstein distance to achieve convergence of parameters.

1. TimeGAN

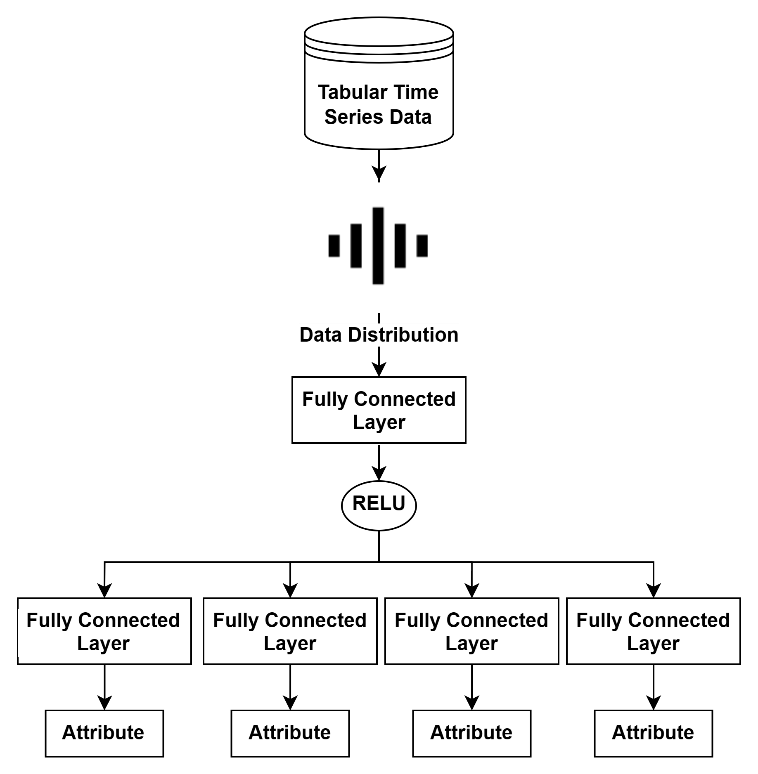
Time-series Generative Adversarial Networks is a GAN-based framework that can produce realistic time-series data in a various domain. TimeGAN introduces the concept of supervised loss, which is different from other GAN architectures (e.g., WGAN) where unsupervised adversarial loss is implemented on both real and synthetic data.

The TimeGAN model is encouraged to capture time conditional distribution within the data by using the original data as a supervision. Additionally, it contains an embedding network, which oversees decreasing the dimensionality of the adversarial learning space.

The TimeGAN utilises three types of loss functions for training - reconstruction loss, which is used in the embedder and recovery part to gauge how well the reconstruction of encoded data is with respect to the original; the supervised loss which helps determine how well the generator is approximating in the latent space; the unsupervised loss that tells the overall loss in terms of the min-max game played by the generator and discriminator.

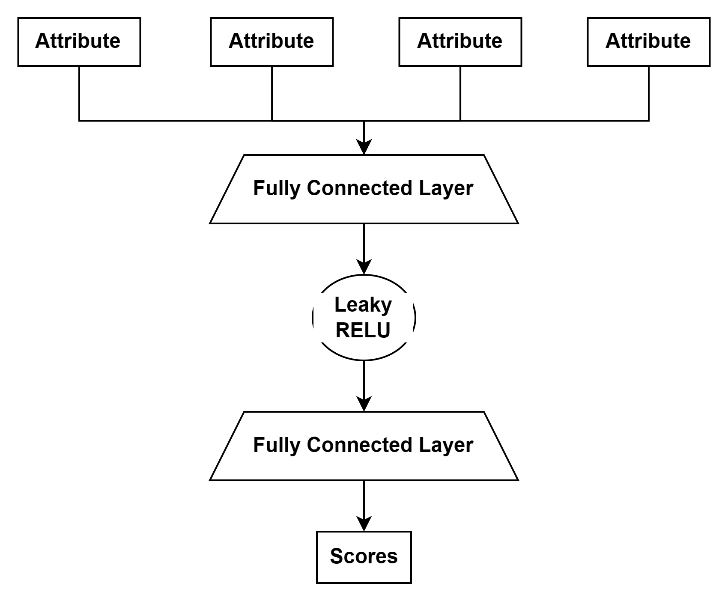
By utilizing the original data as supervision and introducing a stepwise supervised loss in TimeGAN, we explicitly encourage the model to capture the stepwise conditional distributions in the data. This takes advantage of the fact that the training data contain more information than just the real or fake status of each datum; we may specifically learn from the transition dynamics of real sequences. Secondly, to lower the high dimensionality of the adversarial learning space, we construct an embedding network to offer a reversible mapping between features and latent representations. Based on this we design for networks are used: the generator, the discriminator, embedder and the recovery network. Each of these networks are a recurrent neural network which provide their specific functionalities. The discriminator helps capture the stepwise conditional distributions while recovery is present to provide reverse mapping of the features.

1. DP-TWGAN
   1. Generator



**Figure 4.4 Overall Generator Architecture**

* 1. Discriminator



**Figure 4.5 Overall Discriminator Architecture**

1. Long Short-Term Memory

**BIBLIOGRAPHY**

1. Madhu, Madhu, and PAWAN WHIG. "A survey of machine learning and its applications." *International Journal of Machine Learning for Sustainable Development* 4, no. 1 (2022): 11-20.
2. Kapoor, Sayash, and Arvind Narayanan. "Leakage and the reproducibility crisis in ML-based science." *arXiv preprint arXiv:2207.07048* (2022).
3. Liu, Ximeng, Lehui Xie, Yaopeng Wang, Jian Zou, Jinbo Xiong, Zuobin Ying, and Athanasios V. Vasilakos. "Privacy and security issues in deep learning: A survey." *IEEE Access* 9 (2020): 4566-4593.
4. Sun, Hui, Tianqing Zhu, Zhiqiu Zhang, Dawei Jin Xiong, and Wanlei Zhou. "Adversarial Attacks Against Deep Generative Models on Data: A Survey." *arXiv preprint arXiv:2112.00247* (2021).
5. Zhang, Yuheng, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. "The secret revealer: Generative model-inversion attacks against deep neural networks." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 253-261. 2020.
6. Gupta, Rajesh, Sudeep Tanwar, Sudhanshu Tyagi, and Neeraj Kumar. "Machine learning models for secure data analytics: A taxonomy and threat model." *Computer Communications* 153 (2020): 406-440.
7. Raghunathan, Trivellore E. "Synthetic data." *Annual review of statistics and its application* 8 (2021): 129-140.
8. Ghatak, Debolina, and Kouichi Sakurai. "A Survey on Privacy Preserving Synthetic Data Generation and a Discussion on a Privacy-Utility Trade-off Problem." In *Science of Cyber Security-SciSec 2022 Workshops: AI-CryptoSec, TA-BC-NFT, and MathSci-Qsafe 2022, Matsue, Japan, August 10–12, 2022, Revised Selected Papers*, pp. 167-180. Singapore: Springer Nature Singapore, 2023.
9. Ha, Trung, Tran Khanh Dang, Tran Tri Dang, Tuan Anh Truong, and Manh Tuan Nguyen. "Differential privacy in deep learning: an overview." In *2019 International Conference on Advanced Computing and Applications (ACOMP)*, pp. 97-102. IEEE, 2019.
10. Panaretos, Victor M., and Yoav Zemel. "Statistical aspects of Wasserstein distances." *Annual review of statistics and its application* 6 (2019): 405-431.
11. Song, Shuang, Kamalika Chaudhuri, and Anand D. Sarwate. "Stochastic gradient descent with differentially private updates." In *2013 IEEE global conference on signal and information processing*, pp. 245-248. IEEE, 2013.
12. Weng, Lilian. "From gan to wgan." *arXiv preprint arXiv:1904.08994* (2019).
13. Chen, Dingfan, Tribhuvanesh Orekondy, and Mario Fritz. "Gs-wgan: A gradient-sanitized approach for learning differentially private generators." Advances in Neural Information Processing Systems 33 (2020): 12673-12684.
14. Abadi, Martin, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. "Deep learning with differential privacy." In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308-318. 2016.
15. Papernot, Nicolas, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. "Semi-supervised knowledge transfer for deep learning from private training data." arXiv preprint arXiv:1610.05755 (2016).
16. Fan, Liyue. "A survey of differentially private generative adversarial networks." In The AAAI Workshop on Privacy-Preserving Artificial Intelligence, p. 8. 2020.
17. Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." *arXiv preprint arXiv:1411.1784* (2014).
18. Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks." In *International conference on machine learning*, pp. 214-223. PMLR, 2017.
19. Xie, Liyang, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. "Differentially private generative adversarial network." *arXiv preprint arXiv:1802.06739* (2018).
20. Zhang, Xinyang, Shouling Ji, and Ting Wang. "Differentially private releasing via deep generative model (technical report)." *arXiv preprint arXiv:1801.01594* (2018).
21. Xu, Chugui, Ju Ren, Deyu Zhang, Yaoxue Zhang, Zhan Qin, and Kui Ren. "GANobfuscator: Mitigating information leakage under GAN via differential privacy." *IEEE Transactions on Information Forensics and Security* 14, no. 9 (2019): 2358-2371.
22. Jordon, James, Jinsung Yoon, and Mihaela Van Der Schaar. "PATE-GAN: Generating synthetic data with differential privacy guarantees." In *International conference on learning representations*. 2018.
23. Beaulieu-Jones, Brett Kreigh. "Machine Learning Methods to Identify Hidden Phenotypes in the Electronic Health Record." PhD diss., University of Pennsylvania, 2017.
24. Wu, Abraham Noah, Rudi Stouffs, and Filip Biljecki. "Generative Adversarial Networks in the built environment: A comprehensive review of the application of GANs across data types and scales." *Building and Environment* (2022): 109477.
25. Figueira, Alvaro, and Bruno Vaz. "Survey on synthetic data generation, evaluation methods and GANs." *Mathematics* 10, no. 15 (2022): 2733.
26. Ghosheh, Ghadeer, Jin Li, and Tingting Zhu. "A review of Generative Adversarial Networks for Electronic Health Records: applications, evaluation measures and data sources." *arXiv preprint arXiv:2203.07018* (2022).
27. te Marvelde, Pepijn. "Differentially Private GAN for Time Series." (2021).
28. Nieminen, Valtteri. "Differentially private synthetic tabular data generation with a generative adversarial network and privacy amplification by subsampling." (2022).
29. Xie, Liyang, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. "Differentially private generative adversarial network." arXiv preprint arXiv:1802.06739 (2018).
30. Yoon, Jinsung, Daniel Jarrett, and Mihaela Van der Schaar. "Time-series generative adversarial networks." Advances in neural information processing systems 32 (2019).