Generative Adversarial Networks for Synthetic Time Series Data Generation To Monitor Energy Consumption of CNC Machine

Autonomous Multisensor Systems Group
Institute for Intelligent Cooperating Systems
Faculty of Computer Science
Otto von Guericke University, Magdeburg

13.11.2024

Presenters : Archana Yadav, Gowtham Premkumar, Shweta Bambal

Contents

Foundational Concepts of GAN

Introduction

Introduction

Generative Adversarial Networks (GANs) are made up of two neural networks, a **discriminator** and a **generator.** They use adversarial training to produce artificial data that is identical to actual data.

Figure 1: Images generated by a GAN created by NVIDIA.

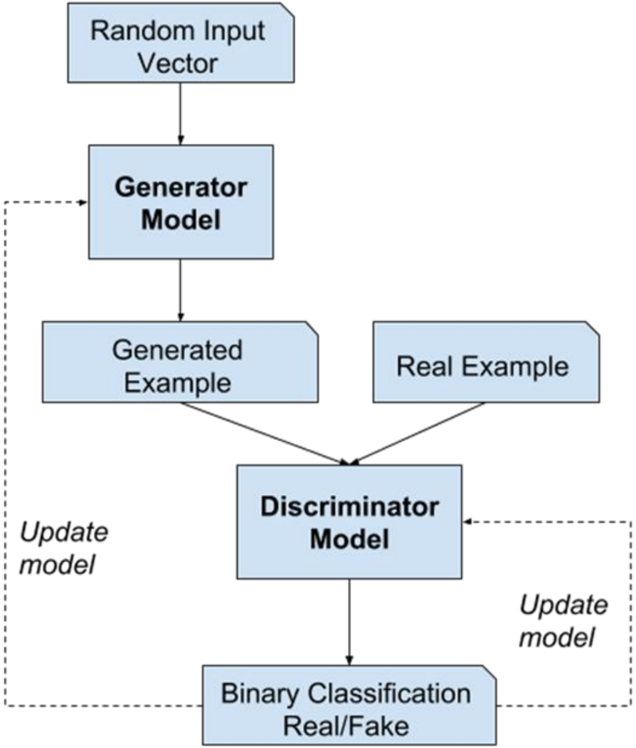


Figure 2: A general GAN architecture

https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/

Understanding the Dataset, Data Preprocessing and Architecture Selection

Project Overview

Overview

Objective: To implement and evaluate different **Generative Adversarial Network** architectures for **Synthetic Time Series Data Generation.**

Step 1:

Understanding the data

Step 2:

Data Preprocessing

Step 3:

Architecture Selection and Training

Step 4:

Evaluation and Results

Understanding the Data

Autonomous Multisensor Systems

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

Name : CNC Machine Data
Size : 18*19393
Data Type: Numerical (all columns)
Attributes: Force Parameters, Moment, Material Removed, Acceleration, Velocity and Position Parameters

f_x_sim	f_y_sim	f_z_sim	f_sp_sim	m_sp_sim	materialremoved_sim	a_x	a_y	a_z	a_sp	v_x	…
-7.74482	192.6447	68.96563	192.8003	0.964002	3.412727	-0.31125	0.065	0.00875	0.85875	-3.5355	…
-7.57726	192.4559	69.05873	192.605	0.963025	3.575455	-0.13125	0.01875	0.0125	3.861875	-3.543	….
-5.73019	190.2727	70.10932	190.359	0.951795	3.487273	0.461875	-0.05375	-0.0075	1.2875	-3.54075	…
-5.96102	190.556	69.97542	190.6492	0.953246	3.55	0.44625	-0.0275	-0.01375	-3.43313	-3.52453	…
…	…	…	…	…	…	…	…	…	…	…	…

Data Preprocessing

Raw Data Loading	Data Reversal	Normalization	Sequence Segmentation	Random Shuffling
• Import raw data from csv file	• Flips data in reverse chronological order • Ensures time series data is in chronological order for sequential analysis	• Min-Max scaling to standardize data to a 0-1 range • Prevents dominance by larger values and improves model convergence.	• Splits data into sequences of fixed length (seq_len). • Prepares sequential input data window	• Shuffles the sequences randomly

Architecture Selection

TimeGAN

RNN based generator and discriminator along with a supervisor.

TransformerGAN

Transformer based generator and discriminator

WGAN

Wasserstein based loss in critic which acts as discriminator

TimeGAN for Time-series Data Generation

TimeGAN

TimeGAN

1. Hybrid Architecture with Embedding Network

2. Multi-objective loss function

3. Supervised Learning Component

4. Adaptability to Multi-Dimensional Time-Series

5. High Fidelity and Improved Training Stability

TimeGAN Architecture

Embedder
Maps the original high - dimensional feature space to a lower - dimensional latent space.

Discriminator
Distinguishes between real and synthetic time-series data.

Recovery
Maps the latent representations back to the original feature space

Supervisor
Teaches the generator the temporal patterns of real data.

Generator
Generates latent space representations from random noise.

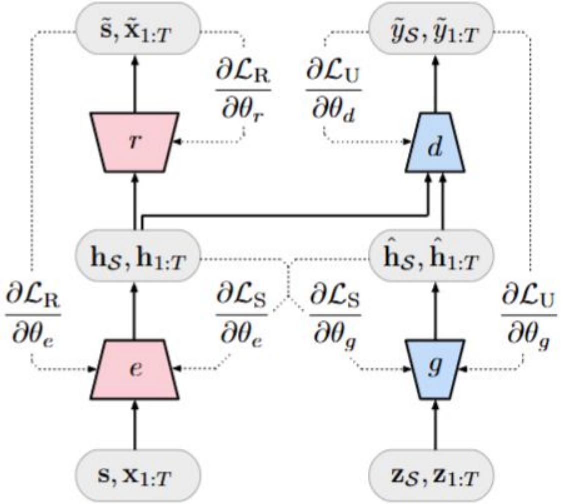


Figure 3: TimeGAN Training Scheme *[1]*

https://www.researchgate.net/publication/344464212_Time-series_Generative_Adversarial_Networks

Transformer GAN for Time-series Data Generation

Transformer GAN

TransformerGAN

1. Long-range Dependencies

2. Self-Attention Mechanism - Identify Complex Patterns

3. Parallel Processing Efficiency

4. Handle Irregular Time-series Data

5. Effective Combination of GANs and Transformers

TransformerGAN Architecture

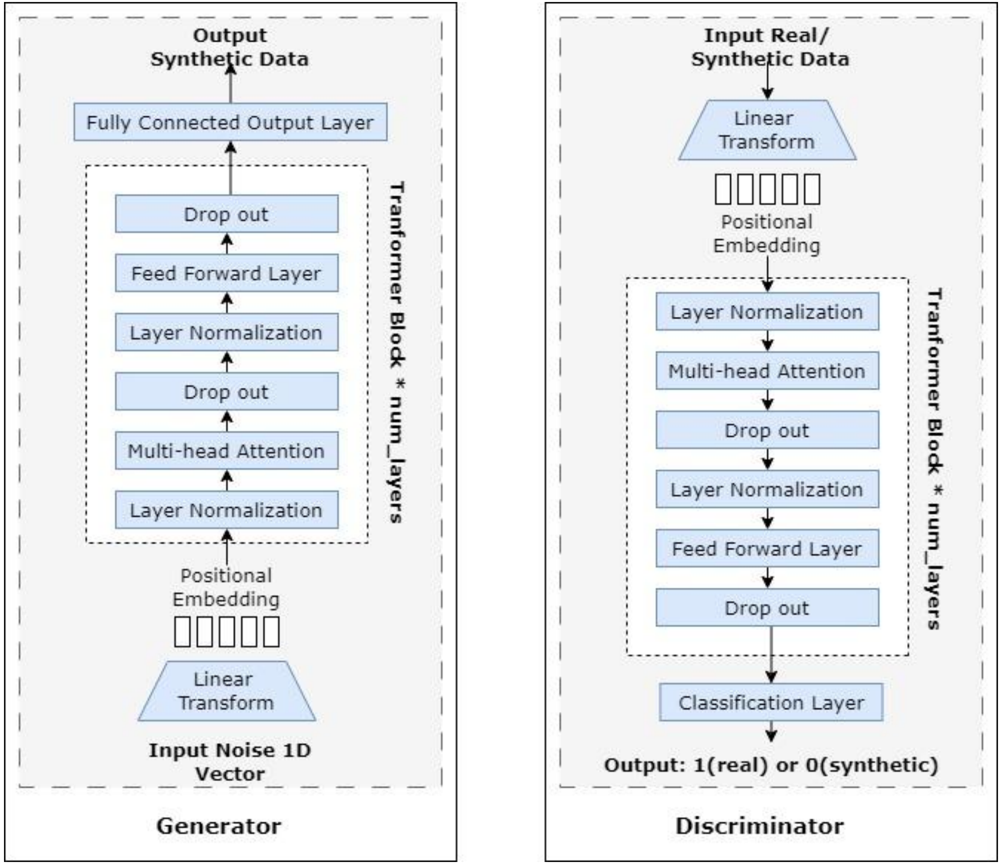


Figure 4 : Transformer GAN Architecture *[2]*

WGAN for Time-series Data Generation

WGAN

WGAN

1. Stabilized GAN Training

2. Improved Loss Function

3. Gradient Clipping for Model Convergence

4.Flexible Architecture for Various Data Types

5. Reduced Sensitivity to Hyperparameters

WGAN Architecture

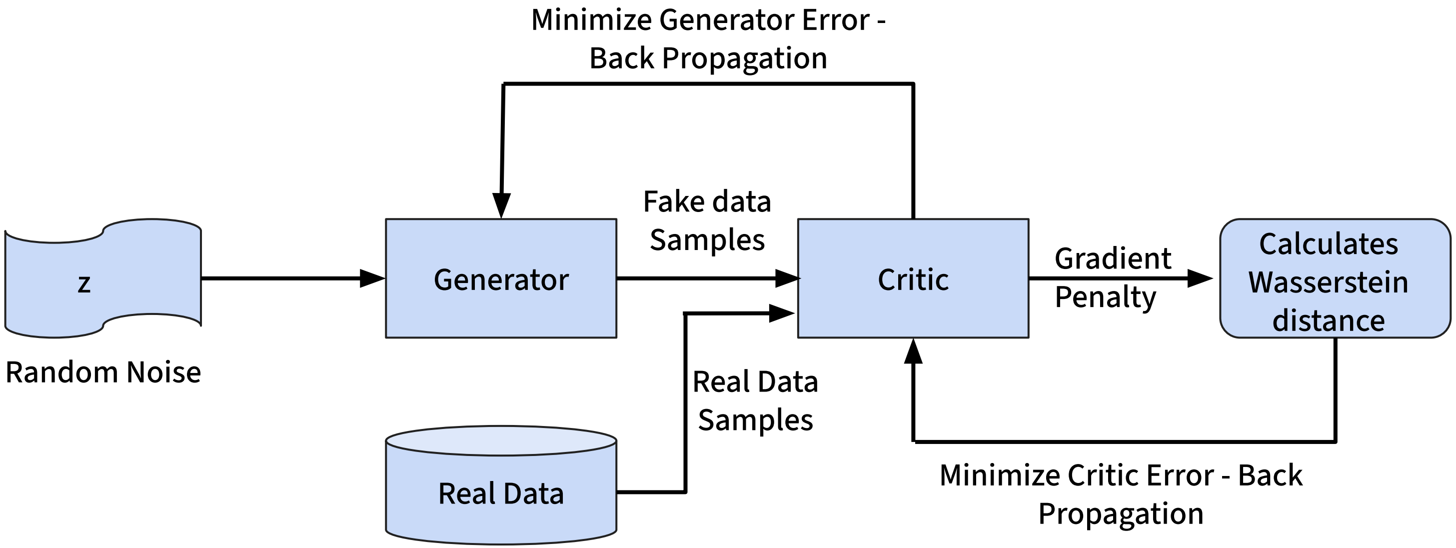


Figure 5 : WGAN Architecture

$$\max_{w\in W} \underbrace{E_{x\sim p_r(x)}[f_w(x)]}_{\text{Expected reward from real data}} - \underbrace{E_{z\sim p(z)}[f_w(g_\theta(z))]}_{\text{Expected reward from generated data}} + \lambda E_{\hat{x}\sim p_{\hat{x}}}[(\|\nabla_{\hat{x}} f_w(\hat{x})\|_2 - 1)^2]$$

[12]

z	*Noise vector sample*
$g_\theta(z)$	*Generator network parameterized by θ*
$f_w(x)$	*Critic network parameterized by w*
x, λ	*Real data sample, Gradient penalty*

Predictive Score, Discriminative Score and FID Score

Evaluation Metrics

Evaluation Metrics

Evaluation metrics ensure that synthetic data closely resembles real data in terms of realism and feature distribution.

Quantitative Analysis

Use of metrics associated with statistical measures used for time series analytics

Qualitative Analysis

Human visual inspection of the generated data.

Evaluation Metrics - Predictive Score

Training on Synthetic Data	Testing on Real Data	Score Calculation
Trained a RNN on synthetic data	Predicted the value of the last feature of the next sequence for the real data using the trained model	Calculated the mean absolute error (MAE) between the predicted and the original feature value of the real data **Average MAE** is the **predictive score**

Evaluation Metrics - Discriminative Score

Test - Train Split

- Labelled real data as 1 and synthetic as 0
- Splitted it into training and testing sets.

Discriminator Training

- Trained discriminator to classify data based on the labels
- Used binary cross entropy loss

Score Calculation

- Calculated the discriminator's accuracy on the test set
- Computed discriminative score as **|accuracy - 0.5|**

Evaluation Metrics - FID Score

Extract Features	Calculate Mean & Covariance	Calculate Frechet Distance

- Used **InceptionTime** neural network
- Capture distributions of the extracted features

- **Mean** - how closely two distributions centered around the same point in feature space.
- **Covariance matrix** - shape and spread of the feature distribution.

- A large **Mean** and **Covariance** difference - synthetic data not same as real data

$$FID(\mu_r, \Sigma_r, \mu_g, \Sigma_g) = \|\mu_r - \mu_g\|_2^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{\frac{1}{2}})$$ [3]

μ_r	*Mean (real)*
μ_g	*Mean (generated)*
Σ_r	*Cov. Matrix (real)*
Σ_g	*Cov. Matrix (generated)*

Evaluation Results of TimeGAN, TransformerGAN and WGAN

Our Results

TimeGAN(LSTM) Results

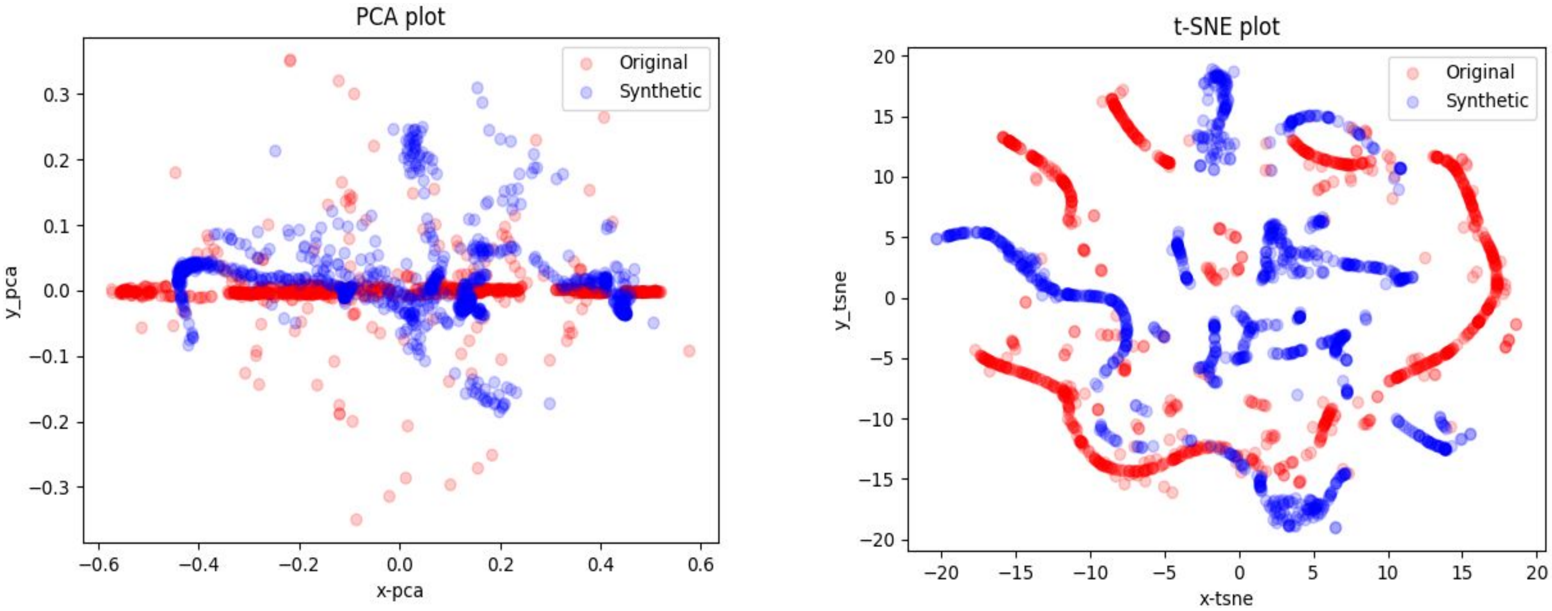


Figure 6 : PCA and t-SNE Plots of TimeGAN(LSTM) for 10k Iterations

TimeGAN(GRU) Results

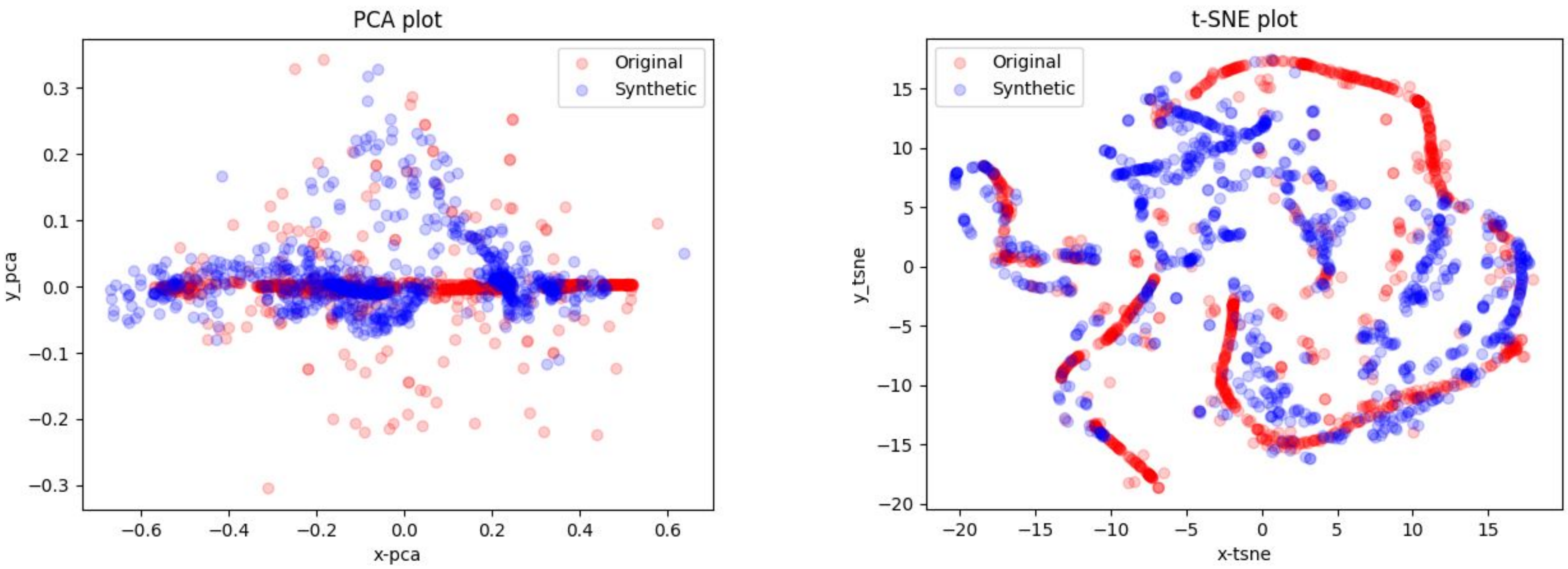
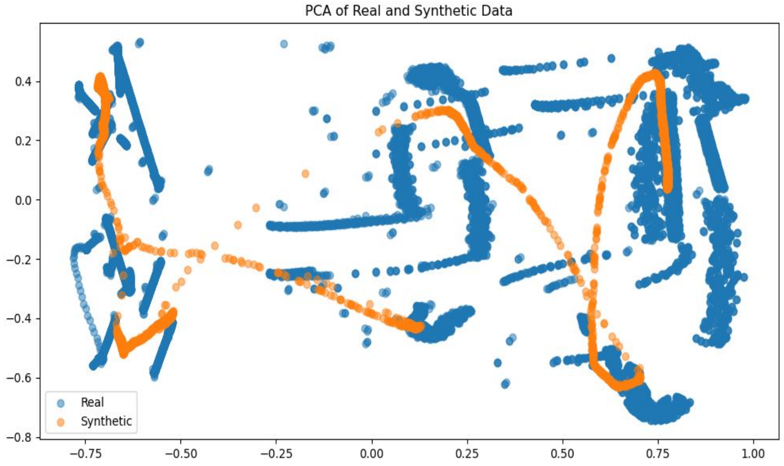


Figure 7 : PCA and t-SNE plots of TimeGAN(GRU) for 10k iterations

TransformerGAN Results



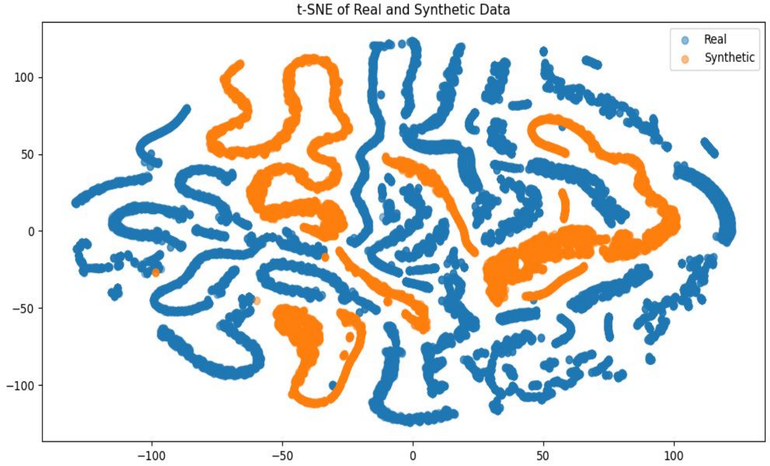
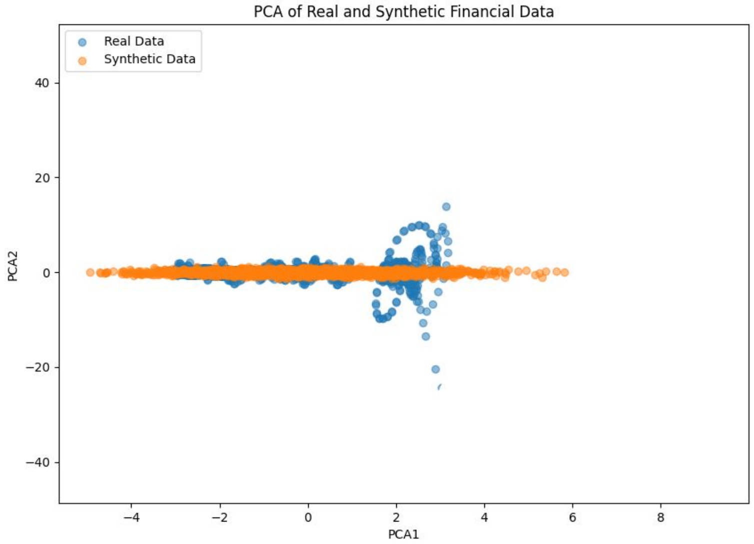


Figure 7 : PCA and t-SNE plots of TransformerGAN for 3k iterations

WGAN Results



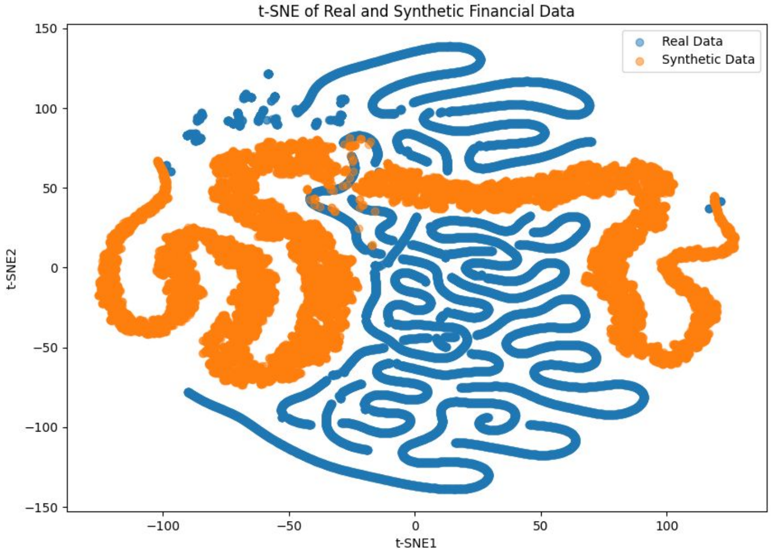


Figure 9 : PCA and t-SNE Plots of WGAN for 10k Iterations

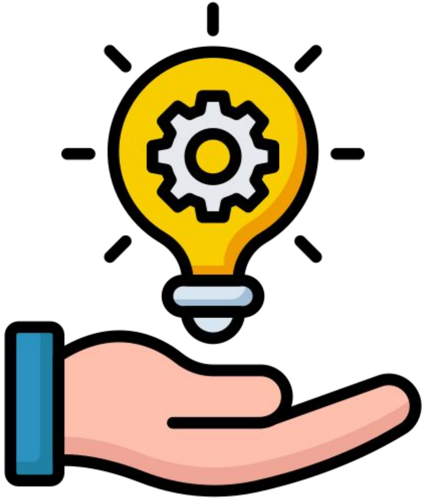
Results

Model	Predictive Score	Discriminative Score	FID Score
TimeGAN (GRU)	0.0111	0.4033	0.6667
TimeGAN (LSTM)	0.0075	0.3634	0.1047
TransformerGAN	0.5024	0.5148	0.2594
WGAN	0.2242	0.5841	0.2867

What our results conclude?

Conclusion

Conclusion



- **TimeGAN (LSTM)** demonstrated the highest performance, effectively generating synthetic data that closely replicates CNC machine energy consumption, showing its value in complex time series applications.

- **Model stability and architecture customization** are key to improving the realism and utility of synthetic data.

Check out our github repo [here]

Opportunities for further Development and Improvement

Future Work

Future Work

- Developing **training techniques** to ensure stable GAN performance and creating **hybrid model architectures** to enhance the quality of synthetic data generation.

- Development of **interactive visualizations** to illustrate the comparison between real data and synthetic data generated by different models.

- **Forecasting energy consumption** patterns of CNC machines by utilizing synthetic data for **energy demand prediction.**

References

1. Yoon, Jinsung & Jarrett, Daniel & Schaar, Mihaela. (2019). Time-series Generative Adversarial Networks. https://www.researchgate.net/publication/344464212_Time-series_Generative_Adversarial_Networks
2. Xiaomin Li, Vangelis Metsis, Huangyingrui Wang, Anne Hee Hiong Ngu. (*2022*) TTS-GAN: A Transformer-based Time-Series Generative Adversarial Network. https://arxiv.org/abs/2202.02691
3. Ricardo de Deijn, Aishwarya Batra, Brandon Koch, Naseef Mansoor and Hema Makkena. (2024). Reviewing FID and SID metrics on Generative Adversarial Networks. https://arxiv.org/pdf/2402.03654
4. Padmanaba Srinivasan, William J. Knottenbelt. (2022) Time-series Transformer Generative Adversarial Networks https://arxiv.org/abs/2205.11164
5. EOIN BROPHY, ZHENGWEI WANG, QI SHE, TOMÁS WARD. (2023) Generative Adversarial Networks in Time Series: A Systematic Literature Review https://dl.acm.org/doi/fullHtml/10.1145/3559540
6. H. Arnout, J. Bronner and T. Runkler, "Evaluation of Generative Adversarial Networks for Time Series Data," 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 2021, pp. 1-7, doi: 10.1109/IJCNN52387.2021.9534373.
7. Brophy, Eoin et al. "Generative adversarial networks in time series: A survey and taxonomy." ArXiv abs/2107.11098 (2021): n. pag.,
8. https://www.techtarget.com/searchenterpriseai/tip/GAN-vs-transformer-models-Comparing-architectures-and-uses
9. https://towardsdatascience.com/evaluation-of-synthetic-time-series-1b4fc4e2be39https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/
10. https://wandb.ai/ayush-thakur/gan-evaluation/reports/How-to-Evaluate-GANs-using-Frechet-Inception-Distance-FID---Vmlldzo0MTAxOTI
11. Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN [arXiv preprint arXiv:1701.07875]. Retrieved from https://arxiv.org/abs/1701.07875
12. Improved Training of Wasserstein GANs by Gulrajani, Ahmed, Arjovsky, Dumoulin, and Courville (2017). https://arxiv.org/abs/1704.00028

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

Generative Adversarial Networks for Time Series

Thank You For Your Attention!

Autonomous Multisensor Systems