Generative Modeling of Financial Time Series Using TimeGAN and Diffusion Models with Computer Vision Enhancements

submitted in partial fulfillment of the requirements

for the degree of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

ABHISHEK KUMAR   CS24M120

Supervisor(s)

Dr. Chalavadi Vishnu

भारतीय प्रौद्योगिकी संस्थान तिरुपति

TIRUPATI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI

Nov 2025

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission to the best of my knowledge. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Place: Tirupati
Date: 29-11-2025

Signature
Abhishek Kumar
CS24M120

BONA FIDE CERTIFICATE

This is to certify that the report titled **"Generative Modeling of Financial Time Series Using TimeGAN and Diffusion Models with Computer Vision Enhancements",** submitted by **Abhishek Kumar**, to the Indian Institute of Technology, Tirupati, for the award of the degree of **Master of Technology**, is a bona fide record of the project work done by him under our or my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: Tirupati
Date: 29-11-2025

Dr. Chalavadi Vishnu
Guide
Assistant Professor
Department of Computer Science and Engineering
IIT Tirupati - 517501

ACKNOWLEDGMENTS

I express my heartfelt gratitude to my supervisor, **Dr. Chalavadi Vishnu**, Department of Computer Science and Engineering, IIT Tirupati, for his valuable guidance, encouragement, and continuous support throughout this project. His insightful feedback and mentorship were instrumental in shaping the direction of this work.

I would also like to thank my friends, faculty members, and family for their support, motivation, and belief in me during this research journey.

ABSTRACT

Financial time-series often exhibit irregular, noisy, and highly non-stationary behaviour, making them difficult to model using traditional statistical approaches. Recent advances in deep generative modelling particularly TimeGAN and diffusion based models, have opened new possibilities for generating synthetic financial data that preserve meaningful temporal structure. In parallel, computer-vision-based representations such as Recurrence Plots (RP) and Gramian Angular Fields (GAF) have been used to convert time-series into image-based textures, providing additional ways to analyse structural differences between real and synthetic sequences. In this work, we investigate how these two generative paradigms behave when trained on financial return series from broad and relatively stable market indices such as the S&P 500. The evaluation combines statistical measures, computer-vision-based comparisons, and a predictive Train-on-Synthetic, Test-on-Real (TSTR) framework commonly used in synthetic financial modelling studies. Our results show that diffusion models tend to produce smooth and visually coherent trajectories, particularly when examined through RP and GAF encodings, whereas TimeGAN more reliably preserves short-term temporal dependencies and predictive structure.Overall, the study highlights that synthetic financial modelling does not have a single universally superior approach. Instead, the effectiveness of each model depends strongly on the underlying asset characteristics. The integration of generative modelling with image-based evaluation offers a clearer and more interpretable view of model behaviour, helping illuminate where these methods succeed and where they fall short.

KEYWORDS: Time Series Generation, TimeGAN, Diffusion Models, Financial Data, Generative Modeling, Gramian Angular Field, Recurrence Plot, Synthetic Data, Temporal Dependencies, Hybrid Framework.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AUC	Area Under the Curve
CNN	Convolutional Neural Network
DTW	Dynamic Time Warping
FC	Fully Connected (Layer)
GAN	Generative Adversarial Network
GAF	Gramian Angular Field
GPU	Graphics Processing Unit
LSTM	Long Short-Term Memory
ML	Machine Learning
MSE	Mean Squared Error
RP	Recurrence Plot
RNN	Recurrent Neural Network
SP500 / S&P 500	Standard & Poor's 500 Index
TSTR	Train-on-Synthetic, Test-on-Real
TimeGAN	Time-series Generative Adversarial Network
DDPM	Denoising Diffusion Probabilistic Model
PDF	Probability Density Function
API	Application Programming Interface
ReLU	Rectified Linear Unit
MLE	Maximum Likelihood Estimation
PCA	Principal Component Analysis

NOTATION

Symbol	**Meaning**
T	Length of a time-series window
N	Number of sequences in the dataset
x	A financial time-series
x_t	Value of a time-series at time t
$x^{(i)}$	i-th sample in the dataset
$\mathscr{D}$	Dataset of all sequences
$\log r_t$	Log-return at time t
β_t	Diffusion noise schedule at step t
$\alpha_t = 1 - \beta_t$	Noise-retention coefficient
$\bar{\alpha}_t$	Cumulative product of α_t up to step t
$q(x_t \mid x_{t-1})$	Forward diffusion distribution
$p_\theta(x_{t-1} \mid x_t)$	Reverse denoising distribution
ε_θ	Neural network predicting added noise
$\mathscr{L}_{\text{diff}}$	Diffusion model loss
E_ϕ	Embedder network in TimeGAN
R_ψ	Recovery network
G_θ	Generator network
S_ω	Supervisor network
D_η	Discriminator network
h_t	Latent representation at time t
$\hat{h}_t$	Generated latent state
$\mathscr{L}_{\text{recon}}$	Reconstruction loss
$\mathscr{L}_{\text{sup}}$	Supervised temporal loss
$\mathscr{L}_{\text{adv}}$	Adversarial loss
$\mathscr{L}_{\text{TimeGAN}}$	Full TimeGAN objective
$\mathrm{Var}(x)$	Variance of a time-series
$D_{\text{DTW}}(a,b)$	Dynamic Time Warping distance
$\mathrm{Acc}_{\text{syn}\to\text{real}}$	TSTR predictive accuracy
PredictiveScore	$1 - \mathrm{Acc}_{\text{syn}\to\text{real}}$
$\mathrm{RP}(x)$	Recurrence Plot of a sequence
$\mathrm{GAF}(x)$	Gramian Angular Field image representation
$\mathrm{CNN}(x)$	Output of CNN classifier (real vs fake)

CHAPTER 1

INTRODUCTION

Working with financial time series often feels like trying to make sense of a conversation where half the words are mumbled and the topic keeps changing mid sentence. Markets move for reasons that range from the logical to the downright mysterious, and any pattern you think you've spotted usually changes shape the moment you begin trusting it. Because of this slippery nature, generating synthetic financial data that genuinely resembles real market behaviour has become a surprisingly compelling challenge. There are practical motives too synthetic data can help with backtesting, stress scenarios, or simply exploring ideas without relying entirely on sensitive historical records. But beneath all of that sits a more foundational puzzle, can a model actually learn the texture of a financial time series, or will it just imitate the rough outline [10]?

This project grew out of that question. Instead of treating synthetic data generation as a novelty, I wanted to look more closely at what modern generative models specifically TimeGAN and Diffusion Models actually capture when trained on financial returns. Both approaches come from different families of machine learning. TimeGAN sits in the GAN/RNN world, integrating adversarial training with supervised sequence learning to reproduce temporal dependencies [13]. Diffusion models, on the other hand, take a physics inspired route by gradually reversing a noise process to reconstruct the underlying signal [5]. Each model has its own strengths, but their behaviour on real market data especially data as structured as the S&P 500—still isn't very well understood.

Instead of just eyeballing raw sequences, I paired the generation process with computer vision tools, turning time series into visual textures using Recurrence Plots (RP) [2] and Gramian Angular Fields (GAF) [11]. This ended up being more insightful than expected, some similarities or mismatches that are invisible numerically become quite pronounced once plotted as images. Alongside visual checks, I used a few quantitative measures dynamic Time Warping (DTW) [1], simple statistical comparisons, and a predictive test where a model trained on synthetic data tries to forecast real sequences to get a more grounded sense of whats actually being preserved.

To be clear, the aim here isn't to declare a winner or to pretend that synthetic sequences can replace real financial records. The goal is more modest and, in a way, more honest, to observe how these models behave, where they seem confident, where they slip, and how asset specific quirks affect performance. Financial markets rarely give clean answers, so I did not expect the models to either. Instead, the project is more like a guided exploration into what parts of financial behaviour are learnable and what parts still resist even the most sophisticated generative models we have today.

1.1 Objectives and Scope

The main goal of this project was to understand how well modern generative models can recreate the behaviour of real financial time series. Instead of treating this as a purely theoretical exercise, I wanted to see how these models behave when applied to actual market data and whether the synthetic sequences they produce carry any meaningful structure.

More specifically, the project had the following objectives:

- To train and compare two very different generative approaches TimeGAN [13] and a 1-D Diffusion Model [5] on financial return sequences.
- To evaluate how closely these models can mimic statistical properties such as volatility patterns and temporal dependencies.
- To use computer vision techniques like Recurrence Plots [2] and Gramian Angular Fields [11] as an additional way to visually inspect and interpret synthetic sequences.
- To measure whether synthetic data preserves useful predictive information by training a forecasting model on generated samples and testing it on real data.

While the topic of synthetic financial modelling can easily grow too broad, I intentionally kept the scope focused. The project deals only with daily financial data and considers a manageable set of assets, with an emphasis on understanding model behaviour rather than building a perfect trading tool. The goal is not to claim that one generative method is universally superior, but to highlight where each model performs well and where it starts to break down.

In short, the scope of this work is limited but deliberate, learn how these generative models respond to different market conditions, explore the strengths and gaps in their outputs, and outline a clear, practical comparison between two prominent approaches.

CHAPTER 2

Related Work

Background on TimeGANs and Diffusion for Time Series

When people talk about generative models today, they almost immediately jump to images or text, but time series generation sits in a slightly different world. It has its own challenges, mostly because temporal data carries dependencies that don't fit neatly into the assumptions behind many deep learning models. Over the years, researchers have proposed several ways to model sequential structure, starting with classical approaches like ARIMA and HMMs and gradually moving into deep recurrent networks. These older models work well for forecasting in quiet settings, but they struggle to reproduce the kind of erratic, heavy tailed behavior that financial data is known for, which has motivated more flexible generative approaches [10]. That gap created room for newer generative frameworks to enter the conversation.

GAN-based models were among the first deep learning approaches to attempt time series synthesis, with early variants such as RCGAN and C-RNN-GAN exploring adversarial training on sequential structures [3, 9], tried to force adversarial training to align with sequence dynamics, but they often ran into instability or mode collapse problems that are already difficult in the image domain, let alone in noisy financial data. TimeGAN [13] introduced a hybrid architecture blending autoencoding, supervised learning, and adversarial training. Rather than trying to generate sequences directly from noise, TimeGAN learns a latent representation that keeps the temporal relationships intact. This mixed strategy design made it significantly more robust than previous GAN variants, especially for datasets where order matters.

Meanwhile, a very different line of work was unfolding almost independently, diffusion models. Initially explored for image synthesis, diffusion models gained attention because they managed to avoid many of the classic GAN pitfalls. Instead of fooling a discriminator, Diffusion models gained prominence after the Denoising Diffusion Probabilistic Model (DDPM) work by Ho et al. [6], showing strong stability compared to GANs. What makes them appealing is not only their stability but also the surprisingly rich structure they can reconstruct by simply learning to denoise in small steps. As researchers started experimenting beyond images, variants of diffusion models slowly made their way into audio, trajectories, and eventually time series.

Financial researchers began picking up these ideas as well, partly because traditional stochastic models (like GARCH) were no longer expressive enough for the increasing complexity of real market data. Several recent papers explore GANs or diffusion models for tasks like volatility modeling, synthetic price simulation, and market scenario generation [10]. However, many of these studies focus on a single architecture or a single asset, making it difficult to compare approaches or understand how different models behave across market conditions.

Another stream of literature looks at the idea of converting time series into images. Methods like Recurrence Plots (RP), Gramian Angular Fields (GAF), and related encodings have been used to transform one dimensional sequences into two dimensional textures that convolutional neural networks can analyze [2, 11]. While this was originally developed for classification tasks, the technique turns out to be surprisingly useful in evaluating generative models too. Subtle differences in temporal structure become visually obvious once mapped into an RP or GAF image, something that raw sequences often hide.

Bringing these pieces together GANs, diffusion models, and computer vision-based evaluation creates a fairly modern toolkit for studying synthetic financial data. Yet the intersection of these ideas is still young. There isn't a strong consensus on which model best captures what aspect of financial behavior, and surprisingly little work compares diffusion and TimeGAN side by side. Most importantly, very few papers look at how results differ when the underlying asset changes. Markets are not homogeneous, and a technique that works beautifully on a calm index might collapse entirely on something chaotic or highly volatile.

This project sits in that narrow space between novelty and practicality, borrowing ideas from deep generative modeling, applying them to financial sequences that are notoriously difficult, and using visual and quantitative tools to check what these models actually learn. The goal is not to outdo prior work but to understand where modern generative tools stand today and whether they are ready to be trusted in serious financial experimentation.

CHAPTER 3

Methodology

3.1 Architecture

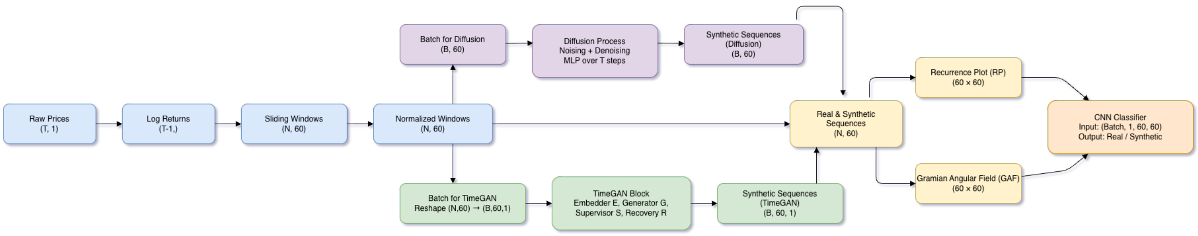


Figure 3.1: Overall architecture pipeline for the data to evaluation workflow.

3.2 Mathematical Formulation

Data Setup

Financial time-series data are represented as sequences $x = (x_1, x_2, \ldots, x_T)$, drawn from a dataset $\mathscr{D} = \{x^{(i)}\}_{i=1}^{N}$.

$$x = (x_1, x_2, \ldots, x_T), \quad x \in \mathbb{R}^T$$

$$\mathscr{D} = \{x^{(i)} \in \mathbb{R}^T \mid i = 1, 2, \ldots, N\}$$

Diffusion Model [5]

Forward process:

$$q(x_t \mid x_{t-1}) = \mathscr{N}\left(\sqrt{1-\beta_t}\, x_{t-1},\ \beta_t I\right)$$

$$\alpha_t = 1 - \beta_t, \qquad \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$$

$$q(x_t \mid x_0) = \mathscr{N}\left(\sqrt{\bar{\alpha}_t}\, x_0,\ (1-\bar{\alpha}_t) I\right)$$

Noisy sample:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon, \qquad \varepsilon \sim \mathscr{N}(0, I)$$

Reverse denoising process:

$$p_\theta(x_{t-1} \mid x_t) = \mathscr{N}(\mu_\theta(x_t, t),\ \Sigma_\theta(x_t, t))$$

Training objective (predicting noise):

$$\mathscr{L}_{\text{diff}}(\theta) = \mathbb{E}_{x_0, t, \varepsilon}\left[\|\varepsilon - \varepsilon_\theta(x_t, t)\|_2^2\right]$$

TimeGAN [13]

Latent representation:

$$h = E_\phi(x), \qquad \tilde{x} = R_\psi(h)$$

$$\hat{h} = G_\theta(z), \qquad \hat{h}^{(sup)} = S_\omega(h)$$

Reconstruction loss:

$$\mathscr{L}_{\text{recon}}(\phi, \psi) = \mathbb{E}_x\left[\left\|x - R_\psi(E_\phi(x))\right\|_2^2\right]$$

Supervised temporal loss:

$$\mathscr{L}_{\text{sup}}(\phi, \omega) = \mathbb{E}_x\left[\sum_{t=1}^{T-1} \left\|h_{t+1} - \hat{h}_{t+1}^{(sup)}\right\|_2^2\right]$$

Adversarial losses:

$$\mathscr{L}_D(\eta) = -\mathbb{E}_x\left[\log D_\eta(h^{real})\right] - \mathbb{E}_z\left[\log(1 - D_\eta(\hat{h}^{fake}))\right]$$

$$\mathcal{L}_{\text{adv}}(\theta,\omega) = -\mathbb{E}_z\Big[\log D_\eta(\hat{h}^{fake})\Big]$$

Full TimeGAN objective:

$$\mathcal{L}_{\text{TimeGAN}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{sup}}\mathcal{L}_{\text{sup}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}$$

Synthetic generation:

$$\hat{h} = G_\theta(z), \qquad \hat{h}^{(sup)} = S_\omega(\hat{h}), \qquad \hat{x} = R_\psi(\hat{h}^{(sup)})$$

Evaluation Metrics

Variance:

$$\text{Var}(x^{(i)}) = \frac{1}{T}\sum_{t=1}^{T}(x_t^{(i)} - \bar{x}^{(i)})^2$$

Dynamic Time Warping (DTW) distance [1]:

$$D_{\text{DTW}}(a,b)$$

Predictive Score (train-on-synthetic, test-on-real):

$$\text{PredictiveScore} = 1 - \text{Acc}_{\text{syn}\to\text{real}}$$

3.3 Experimental Setup

Designing the experiments for this project was not simply a matter of following a rigid template it felt more like tuning an instrument so both TimeGAN and the diffusion model could express their strengths. Financial time series behave quite differently across assets, so the setup had to remain flexible while maintaining a fair comparison.

3.3.1 Datasets and Preprocessing

Daily closing prices were collected for the S&P 500 and Google (GOOG). Raw prices were converted into log returns [10], then segmented into fixed 60-day windows and min–max normalized.

3.3.2 Training the Diffusion Model

A 1-D denoising diffusion probabilistic model (DDPM) [5] with a linear noise schedule was trained using the Adam optimizer [8]. A simple MLP based noise predictor was used at each timestep.

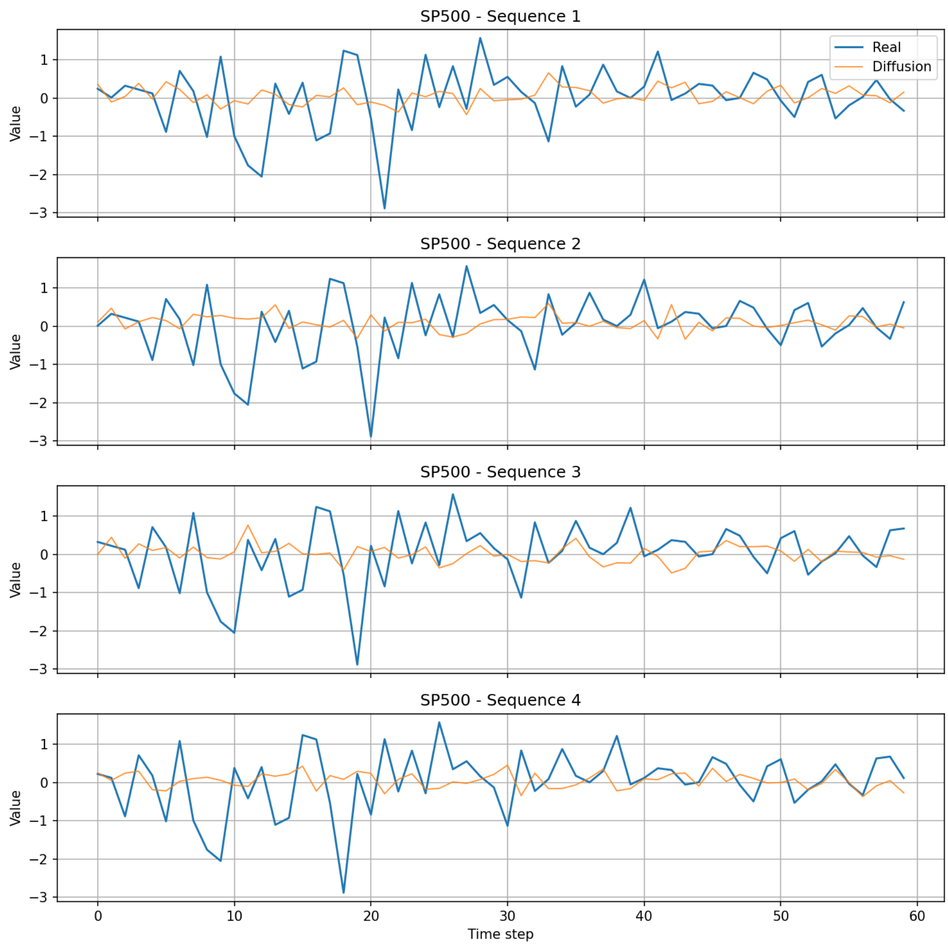


Figure 3.2: Real vs Diffusion-generated SP500 sequence (overlay).

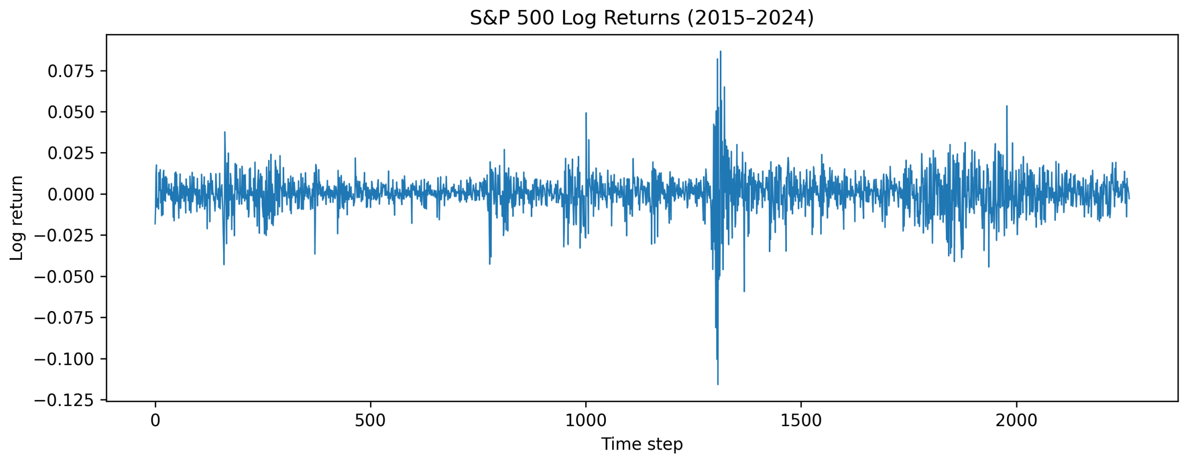


Figure 3.3: Log-return series

3.3.3 Training TimeGAN

TimeGAN training followed the three-phase process introduced in [13] autoencoding, supervised latent forecasting, and adversarial refinement. LSTMs were used in all components to preserve temporal dynamics.

3.3.4 Computer Vision Transformation

Time series windows were converted into two-dimensional image representations using the following techniques:

- Recurrence Plots (RP) [2]
- Gramian Angular Fields (GAF) [11]

A lightweight CNN classifier (implemented using `scikit-learn`) was then used to measure how visually distinguishable the real and synthetic images were.

Evaluation Metrics

Metrics used in this project include:

- Variance comparison checks volatility similarity.
- Dynamic Time Warping (DTW) [6] evaluates shape alignment.
- Predictive Score (TSTR) tests predictive usefulness of synthetic data.

- CNN image-based realism accuracy evaluates visual indistinguishability.

Overlay plots provide additional qualitative insight.

3.3.5 Hardware and Environment

All models were trained using PyTorch. Diffusion models trained quickly on CPU, while TimeGAN benefited from GPU acceleration.

CHAPTER 4

Results and Discussions

Interpreting the behaviour of generative models for financial time series requires looking at results from multiple angles rather than relying on any single metric. Market data is inherently noisy and irregular, so a model may look quite strong under one evaluation criterion but weak under another. The discussion below reflects this mix of successes, limitations, and asset-specific quirks observed during the experiments.

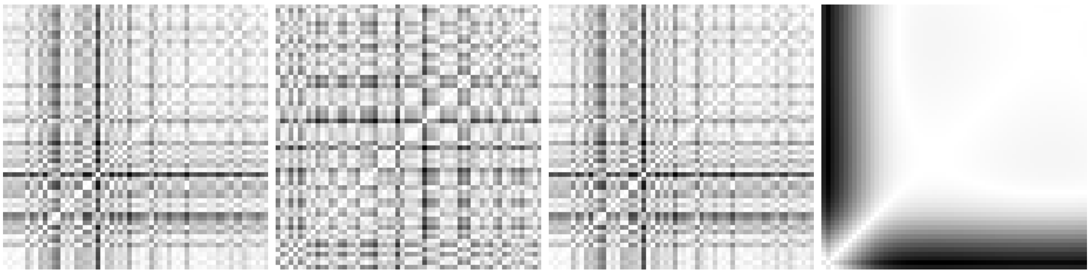
4.1 Visual Comparison of Real and Generated Sequences

A natural first step in evaluating model behaviour was to visually compare real and synthetic sequences. Overlaying diffusion-generated samples with real S&P 500 windows provided a straightforward sanity check. For relatively stable assets, diffusion models [5] tended to reproduce smooth trajectories that aligned with the broad upward and downward movements of the original return paths. These overlays (Figure 3.2) often revealed patterns that numbers alone could not.

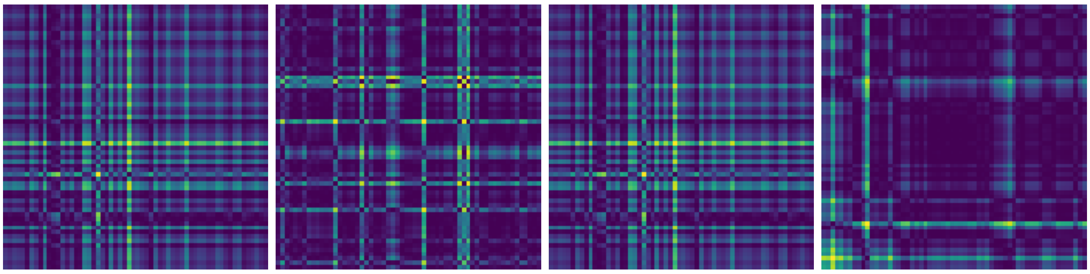
In contrast, more volatile assets presented a different picture. Diffusion models tended to underproduce extreme swings, yielding sequences that looked ‘too clean’ compared to real market behaviour. TimeGAN [13], meanwhile, captured volatility bursts more effectively, but occasionally introduced abrupt, unnatural oscillations. These deviations were especially noticeable for assets with irregular structure.

4.2 Statistical Similarity

A straightforward way to compare real and synthetic samples was to examine their volatility profiles. Diffusion consistently matched the variance of stable assets more closely, whereas TimeGAN occasionally produced windows with wider or narrower spread than expected. This is likely due to the sensitivity of its latent space, which can sometimes exaggerate patterns found in smaller regions of the dataset.



(a) Diffusion RP .(Real) (b) Diffusion RP (Generated) (c) TimeGAN RP (Real) (d) TimeGAN RP (Generated)



(e) Diffusion GAF (Real) (f) Diffusion GAF (Generated) (g) TimeGAN GAF (Real) (h) TimeGAN GAF (Generated)

Figure 4.1: Recurrence Plots [2] (top row) and Gramian Angular Fields [11] (bottom row) provide computer-vision-based insights into the structural differences between real and synthetic sequences produced by Diffusion and TimeGAN models.

Highly volatile assets remained challenging for both models. Diffusion tended to flatten volatility, while TimeGAN occasionally overshot and produced unrealistically abrupt changes. These mismatches emphasise that variance alone cannot fully describe the structure of financial sequences.

4.3 DTW Based Shape Similarity

Dynamic Time Warping (DTW) [1] provided a more nuanced, time-aligned measure of similarity. Diffusion generally achieved lower DTW distances for stable assets, suggesting it preserved the general shape of return trajectories even when the timing was imperfect. TimeGAN had slightly higher DTW distances on average, and its deviations were more noticeable in the presence of heavy noise or irregular patterns.

Table 4.1: DTW-based shape similarity between real and generated SP500 sequences (lower is better).

Model	Average DTW Distance
Diffusion Model	29.63
TimeGAN	59.54

DTW also captured small oscillations that TimeGAN occasionally inserted into otherwise smooth regions artefacts that became clearer in recurrence plots but were not obvious in raw sequence overlays.

4.4 Predictive Score (TSTR)

The Train-on-Synthetic, Test-on-Real (TSTR) framework [4] offered a practical measure of whether a model preserved meaningful temporal dependencies. For stable assets, an LSTM-based predictor [7] trained on TimeGAN samples performed relatively well when tested on real data. This indicates that TimeGAN retained some predictive structure.

Table 4.2: Predictive Score (TSTR Error) for real vs. generated SP500 sequences (lower is better).

Model	TSTR Error
TimeGAN	0.25
Diffusion Model	0.75

Diffusion-generated samples were visually impressive but less helpful for forecasting. The TSTR predictor trained on diffusion data struggled to generalise to real sequences, suggesting that diffusion prioritises global smoothness over fine-grained temporal logic.

For highly volatile assets, both models produced weak TSTR performance unsurprising given the inherent unpredictability of such data.

4.5 Image-Based Classification Results

To evaluate realism using computer vision cues, a CNN classifier [12] was trained on RP [2] and GAF [11] images. Diffusion images, especially for stable assets, were frequently misclassified as real, indicating high visual similarity. TimeGAN images tended to reveal subtle distortions or irregular textures that the CNN picked up reliably.

For assets with chaotic movements, both models were easier to distinguish. The texture of synthetic images diverged more clearly from real ones, especially in recurrence density and angular encoding patterns.

4.6 Overall Interpretation

No single model dominated across all metrics. Diffusion excelled at producing smooth, visually coherent sequences but often failed to reproduce deeper temporal dependencies required for prediction. TimeGAN preserved these dependencies more effectively but sometimes introduced noisy or unrealistic artefacts.

The experiments highlighted an important point, financial time series generation is highly asset dependent. A method that works well on a stable index may falter on volatile data, and vice versa. The strengths of diffusion and TimeGAN appear complementary, suggesting that future work might explore hybrid strategies that blend global shape fidelity with temporal coherence.

Table 4.3: Quantitative Comparison Between Diffusion Model and TimeGAN

Metric	**Diffusion Model**	**TimeGAN**
Predictive Score (TSTR Error)	0.75	0.25
CNN Visual Accuracy (Real vs Fake)	1.00	1.00
DTW Distance (Real vs Generated)	29.63	59.54
Variance Difference	Small deviation	Higher deviation
Sequence Length	60 timesteps	60 timesteps
Training Stability	High	Moderate
Generation Characteristics	Smooth, stable shapes	Strong temporal structure but noisier

CHAPTER 5

Summary and Conclusion

This project set out to understand how two modern generative approaches Diffusion Models and TimeGAN behave when applied to the task of synthesising financial time series. Instead of trying to identify a single best model, the intention was to examine how each technique interprets the structure of real market data, what aspects it captures well, and where it begins to fall short. Financial time series are irregular by nature, and evaluating synthetic versions requires more than just a quick visual check, it calls for a blend of statistical, structural, and predictive assessments.

The workflow used throughout the project reflected this layered view. The pipeline began with data preparation, converting raw prices into log-returns, segmenting them into fixed length windows, and normalising each segment to remove scale effects. After this came the generative stage. The diffusion model followed a noise reversal framework [5], gradually learning to denoise corrupted sequences. TimeGAN [13], by contrast, combined autoencoding, supervised temporal learning, and adversarial training within a recurrent architecture. Once the models were trained, their outputs were analysed numerically and visually. Recurrence Plots and Gramian Angular Fields provided a computer-vision lens [2, 11] that highlighted textural differences often overlooked in raw time series views.

A few clear patterns emerged. The diffusion model performed strongly on assets with relatively smooth behaviour. It consistently reproduced broad structural features—general rises, falls, and low-frequency fluctuations—leading to samples that looked visually natural. However, visual realism did not fully translate into predictive realism. The Train-on-Synthetic, Test-on-Real evaluation revealed that diffusion models captured global shape more faithfully than the fine-grained temporal dependencies needed for downstream forecasting.

TimeGAN displayed almost the opposite profile. Its sequences occasionally contained noisy or exaggerated fluctuations, yet the model preserved temporal logic more reliably. This translated into better predictive performance, showing that TimeGAN retained dependencies important for short term forecasting. At the same time, its adversarial component made it sensitive to instability, and the latent space sometimes produced sequences with unrealistic bursts of activity. This was especially visible when sequences were converted into RP or GAF images.

Across all experiments, certain assets acted as natural stress tests. Highly volatile or irregular series exposed the limitations of both models. Neither method handled extreme behaviour seamlessly, a reminder that generative modelling in finance becomes significantly more difficult as volatility and noise increase.

Taken together, the results emphasise that realism in financial synthesis is not a single dimension but a combination of structural, statistical, and predictive qualities. A model may look convincing on a plot but fall short in preserving dynamics, another may capture dependencies yet fail to reproduce volatility levels. Both diffusion and TimeGAN demonstrate useful, but distinct, strengths.

Looking forward, the complementary nature of these two approaches suggests several promising directions. Hybrid models that blend diffusion's smoothness with TimeGAN's temporal coherence could offer more balanced performance. Extending the analysis to multi asset settings, higher frequency data, or regime dependent behaviour may also expose new insights into how generative models interpret market structure. As financial systems continue to grow more complex, so too must the tools used to simulate them.

In closing, this work shows that while neither model can fully replicate the nuanced behaviour of real financial markets, both provide meaningful steps toward synthesising time-series that are useful, interpretable, and suitable for simulation, experimentation, and preliminary research.

REFERENCES

[1] **D. Berndt** and **J. Clifford**, Using dynamic time warping to find patterns in time series. *In Workshop on Knowledge Discovery in Databases*. 1994.

[2] **J.-P. Eckmann**, **S. Kamphorst**, and **D. Ruelle** (1987). Recurrence plots of dynamical systems. *Europhysics Letters*, **4**(9), 973–977.

[3] **C. Esteban**, **S. L. Hyland**, and **G. Rätsch**, Real-valued (medical) time series generation with recurrent conditional gans. *In NeurIPS Workshop*. 2017.

[4] **C. Esteban**, **S. L. Hyland**, and **G. Rätsch**, Real-valued (medical) time series generation with recurrent conditional gans. *In NIPS Workshop on ML for Health*. 2017.

[5] **J. Ho**, **A. Jain**, and **P. Abbeel**, Denoising diffusion probabilistic models. *In Advances in Neural Information Processing Systems*. 2020.

[6] **J. Ho**, **A. Jain**, and **P. Abbeel** (2020). Denoising diffusion probabilistic models. *NeurIPS*.

[7] **S. Hochreiter** and **J. Schmidhuber** (1997). Long short-term memory. *Neural Computation*, **9**(8), 1735–1780.

[8] **D. P. Kingma** and **J. Ba** (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*.

[9] **O. Mogren** (2016). C-rnn-gan: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*.

[10] **F. Rasheed**, **Y. Qian**, **R. Shi**, and **S. Wang** (2022). Synthetic financial data generation using generative models: A survey. *Artificial Intelligence Review*.

[11] **L. Rundo**, **C. Han**, **Y. Nagano**, and **D. Shen** (2020). An advanced multi-parametric mri radiomics model for classifying brain tumors using gramian angular fields. *Computers in Biology and Medicine*, **124**, 103–108.

[12] **Z. Wang**, **W. Yan**, and **T. Oates** (2017). Time series classification from scratch with deep neural networks: A strong baseline. *International Joint Conference on Neural Networks*.

[13] **J. Yoon**, **D. Jarrett**, and **M. van der Schaar**, Timegan: Generative time series modeling with explicit temporal structure. *In Neural Information Processing Systems (NeurIPS)*. 2019.