A Generative Adversarial Graph Neural Network for Synthetic Time Series Data

Marco Gregnanin[1,2,*,†], Johannes De Smedt[2,†], Giorgio Gnecco[1,†] and Maurizio Parton[3,†]

[1] *Laboratory for the Analysis of compleX Economic Systems (AXES), IMT School for Advanced Studies Lucca, Piazza S. Ponziano, 6, Lucca, 55100, Tuscany, Italy*

[2] *Research Centre for Information Systems Engineering (LIRIS), KU Leuven, Naamsestraat 69, Leuven, 3000, Flemish Region, Belgium*

[3] *Department of Economic Studies, University of Chieti–Pescara, Viale Pindaro 42, Pescara, 65127, Abruzzo, Italy*

Abstract

Generating synthetic data for financial time series poses challenges, especially taking into account their non-stationary nature. In this work, we introduce the Sig-Graph Generative Adversarial Network (GAN) model, which integrates the following three components: the time series signature, offering a structured summary of temporal evolution of a times series; a Long Short-Term Memory (LSTM) network, capturing its inherent autoregressive structure; and Graph Neural Networks (GNNs), leveraging geometric patterns within the time series data. Numerical evaluation demonstrates that the Sig-Graph GAN model outperforms several baseline models in replicating the distribution of logarithmic returns over the Standard and Poor's 500 stock exchanges.

Keywords

Graph Neural Networks, Signature Transform, Synthetic Time Series

1. Introduction

Across various domains including, among others, economics and finance, the necessity arises for the generation of synthetic data. This need is driven by several factors, such as the scarcity of original data due to privacy concerns and the requirement for data diversity to enhance model generalization. However, the generation of synthetic time series data poses a considerable challenge due to their stochastic nature. This holds especially true for the financial case. We show that geometric patterns play an important role in addressing the generation of synthetic data for a time series, based on a Generative Adversarial Network (GAN) model. Specifically, transforming time series from a Euclidean to a non-Euclidean space using a graph-based approach can significantly enhance our understanding and analysis of complex financial time series behavior. Furthermore, the adoption of a graph-based representation of a time series liberates one from the constraint of assuming stationarity within the time series, and facilitates the analysis of geometric patterns (in the now graph-based representation of the time series) through Graph Neural Networks (GNNs) [1]. Furthermore, we exploit a Long Short-Term Memory (LSTM) network to deal with temporal and long-term patterns. Finally, we explore the application of the time series signature [2], which is a concept derived from path theory. The signature can be viewed as analogous to the Moment Generating Function (MGF), which is useful for comparing random variable distributions as it encodes all distribution moments into a single function.

Discovery Science - Late Breaking Contributions 2024

[*] Corresponding author.

[†] These authors contributed equally.

marco.gregnanin@imtlucca.it (M. Gregnanin); johannes.desmedt@kuleuven.be (J. D. Smedt); giorgio.gnecco@imtlucca.it (G. Gnecco); maurizio.parton@unich.it (M. Parton)

0009-0004-3205-6997 (M. Gregnanin); 0000-0003-0389-0275 (J. D. Smedt); 0000-0002-5427-4328 (G. Gnecco); 0000-0003-4905-3544 (M. Parton)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-3928/paper_137.pdf

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

2. Problem Formulation

Consider a univariate time series $S = \{s_1, s_2, \ldots, s_T\}$ extending up to time T. The objective of this research is to determine a function $f(\cdot)$, given a random variable Z and a graph-based representation G of the original time series S, that is able to generate synthetic data S'. The synthetic data should closely resemble the statistical characteristics, temporal dependencies, and geometric patterns observed in the original time series S. Therefore, we want to find a function $f(\cdot)$ such that $\{s_1, \ldots, s_T\} \overset{d}{\simeq} \{s'_1, \ldots s'_T\}$, where $S' = f(Z, G)$.

3. Proposed Sig-Graph GAN Framework

The initialization involves the construction of the time series $S_t = \{s_{t-m}, \ldots, s_t\}$ observed in $\tilde{T} = m + 1$ instants, and the random noise matrix $Z_t \in \mathbb{R}^{\tilde{T} \times F}$, where F is the dimension of the noise vector. Subsequently, the corresponding graph G_t associated with the time series is formed utilizing the visibility graph algorithm [3]. The choice between an undirected or directed graph is considered a hyperparameter to optimize. Regardless, the adjacency matrix A_t maintains of dimension $\tilde{T} \times \tilde{T}$, with each node corresponding to a specific time observation. The discriminator/generator functions of the GAN model are structured, respectively, as: $Dis(S_t, \eta, A_t)$, and $Gen(Z_t, \theta, A_t)$. Here, η and θ represent vectors of learnable parameters for the discriminator and the generator, respectively. We adopt an identical network configuration for both the generator and the discriminator, and the network structure combines GNN, LSTM, and Fully Connected (FC) layers.

Recurrent Block. The recurrent block processes an input denoted as Z_t for the generator and S_t for the discriminator agent. For simplification, we represent this input as $X_t \in \mathbb{R}^{\tilde{T} \times F}$, where F is set to 1 for the discriminator. The input goes through LSTM layers, serving to capture temporal and long-term patterns. Subsequently, a fully connected layer concludes the recurrent block, producing $\hat{X}_t^1 \in \mathbb{R}^{\tilde{T} \times F}$ as output.

Geometric Block. The geometric block accepts input $X_t \in \mathbb{R}^{\tilde{T} \times F}$ along with the adjacency matrix $A_t \in \mathbb{R}^{\tilde{T} \times \tilde{T}}$. In our model, the Graph Convolution Network (GCN) [4] is used as GNN for analyzing the geometric patterns within the time series data. Subsequently, an LSTM layer processes the data, facilitating the treatment of temporal and long-term patterns uncovered by the GNN layers. Then, after the application of a fully connected layer, the geometric block yields an output $\hat{X}_t^2 \in \mathbb{R}^{\tilde{T} \times F}$.

Linear Block. Within the linear block, the outputs from both the recurrent and geometric blocks, $\hat{X}_t^1 \in \mathbb{R}^{\tilde{T} \times F}$ and $\hat{X}_t^2 \in \mathbb{R}^{\tilde{T} \times F}$ respectively, are processed. The initial step involves summing the two outputs, followed by concatenation with the initial input $X_t \in \mathbb{R}^{\tilde{T} \times F}$. Subsequently, three fully connected layers with unit counts of 128, 64, and 1 are applied. The final output of each network is denoted as $\hat{S}_t^*$, wherein $*$ is replaced with "*real*" for the discriminator and "*fake*" for the generator.

3.1. Loss Function

Prior to computing the loss function, we subject $\hat{S}_t^*$ to the lead-lag transformation, denoted with $L(\cdot)$. To this end, we apply the truncated signature with a truncation level degree set to 5. For more comprehensive insights, we also perform a cumulative summation on $\hat{S}_t^*$, subsequently leading to the computation of a cumulative truncated signature [5]. The custom loss function adopted is the Mean Squared Error (MSE). Denoting the truncated signature as $\mathcal{S}_M(\cdot)$ and the cumulative truncated signature as $\mathcal{S}_M^C(\cdot)$ (both having the same dimension N), the loss function is defined as follows:

$$
\begin{aligned}
\mathrm{MSE}(\hat{S}_t^f, \hat{S}_t^r) = \frac{1}{N} \sum_{i=1}^{N} \left(\mathcal{S}_M(L(\hat{S}_t^f))_i - \mathcal{S}_M(L(\hat{S}_t^r))_i \right)^2 \\
+ \frac{1}{N} \sum_{i=1}^{N} \left(\mathcal{S}_M^C(L(\hat{S}_t^f))_i - \mathcal{S}_M^C(L(\hat{S}_t^r))_i \right)^2 ,
\end{aligned}
$$

with "*f*" and "*r*" denoting respectively the "*fake*" and "*real*" data.

4. Experimental Evaluation

We consider, as baseline models, the Quant GAN model, the GARCH(1, 1) model [6], and a Monte-Carlo simulation for the Black and Scholes model.

Dataset and Pre-Processing. For the scope of our analysis, we selected the Standard & Poor's 500 (S&P 500) stock exchanges. We collected the closing prices of these stock exchanges spanning the interval from January 4, 2010, to December 30, 2019. Each dataset consists of about 2515 observations. Before subjecting the dataset to normalization to achieve a mean of zero and a variance of one, a preliminary step involved computing logarithmic returns denoted as $r_t = \log(s_t) - \log(s_{t-1})$. We chose $\tilde{T} = 100$, and $F = 4$ for the generator.

Evaluation Metrics. To facilitate meaningful comparisons among different models, we employ various evaluation metrics. Notably, we utilize the leverage effect score as defined in [7], and a distribution-based metric. In particular, we consider the Earth Mover's Distance (EMD), also known as the Wasserstein 1-distance [8]. This metric quantifies the minimal cost required to transform the distribution of real data into that of generated data. Then, we compute the Root Mean Squared Error (RMSE) between the signatures of real and generated data.

Numerical Results. Data generation is conducted at different temporal intervals: daily, weekly, monthly, and long-term, corresponding to 1, 5, 20, and 100 days, respectively. Results for both real and generated data for the various datasets are presented in Table 1. Optimal results are highlighted in bold. Our proposed model consistently outperforms the baseline models in terms of the EMD and leverage effect metric.

Evaluation metric	QuantGAN	Sig-Graph GAN(MSE)	MC	GARCH(1,1)
EMD(1)	0.1483	**0.1274**	50.3969	68.8657
EMD(5)	0.3681	**0.3144**	252.3970	151.0271
EMD(20)	1.0954	**0.8679**	1009.8142	305.4143
EMD(100)	4.2506	**4.0865**	5034.7390	695.6773
Sig-RMSE(1)	4.1200	4.0911	**4.0809**	5618.0861
Sig-RMSE(5)	4.2529	4.2228	**4.2123**	5087.6136
Sig-RMSE(20)	3.9921	3.9763	**3.9120**	6263.2784
Sig-RMSE(100)	4.6748	4.6810	**4.5633**	6461.6048
Leverage Effect	3.8231	3.9510	4.125	**3.9218**

Table 1
Results for the real and the generated data for the Standard & Poor's 500 (S&P500) datasets. To facilitate results comparison, the values are multiplied by 100.

5. Conclusion

We introduce a novel approach that combines GNN, LSTM networks, and the Signature transformation to construct a GAN model for the generation of synthetic stock log-returns. Our methodology leverages the inherent geometric patterns present within the time series data. We demonstrate that our proposed model consistently surpasses baseline models. Future works could involve extending the Sig-Graph GAN model to tackle other time series generation challenges (also in contexts different from finance), as well as assessing its potential in enhancing the performance of trading strategies based on synthetic data.

Acknowledgments

Marco Gregnanin and Giorgio Gnecco were partially supported by the PRIN PNRR 2022 project "MOTUS" (CUP: D53D23017470001), funded by the European Union – Next Generation EU program.

References

[1] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Transactions on Neural Networks 20 (2008) 61–80.

[2] T. J. Lyons, Differential equations driven by rough signals, Revista Matemática Iberoamericana 14 (1998) 215–310.

[3] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, J. C. Nuno, From time series to complex networks: The visibility graph, Proceedings of the National Academy of Sciences 105 (2008) 4972–4975.

[4] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016. URL: https://arxiv.org/pdf/1609.02907. arXiv:1609.02907.

[5] I. Chevyrev, A. Kormilitzin, A primer on the signature method in machine learning, 2016. URL: https://arxiv.org/pdf/1603.03788. arXiv:1603.03788.

[6] T. Bollerslev, Generalized autoregressive conditional heteroskedasticity, Journal of Econometrics 31 (1986) 307–327.

[7] M. Wiese, R. Knobloch, R. Korn, P. Kretschmer, Quant GANs: Deep generation of financial time series, Quantitative Finance 20 (2020) 1419–1440.

[8] Y. Rubner, C. Tomasi, L. J. Guibas, The earth mover's distance as a metric for image retrieval, International Journal of Computer Vision 40 (2000) 99–121.