electronics

MDPI

Article

Enhancing Financial Time Series Prediction with Quantum-Enhanced Synthetic Data Generation: A Case Study on the S&P 500 Using a Quantum Wasserstein Generative Adversarial Network Approach with a Gradient Penalty

Filippo Orlandi , Enrico Barbierato * and Alice Gatti

Department of Mathematics and Physics, Catholic University of the Sacred Heart, 25121 Brescia, Italy; filippo.orlandi01@icatt.it (F.O.); alice.gatti@unicatt.it (A.G.)
* Correspondence: enrico.barbierato@unicatt.it

Abstract: This study introduces a novel Quantum Wasserstein Generative Adversarial Network approach with a Gradient Penalty (QWGAN-GP) model that leverages a quantum generator alongside a classical discriminator to synthetically generate time series data. This approach aims to accurately replicate the statistical properties of the S&P 500 index. The synthetic data generated by this model were compared to the original series using various metrics, including Wasserstein distance, Dynamic Time Warping (DTW) distance, and entropy measures, among others. The outcomes demonstrate the model's robustness, with the generated data exhibiting a high degree of fidelity to the statistical characteristics of the original data. Additionally, this study explores the applicability of the synthetic time series in enhancing prediction models. An LSTM (Long-Short Term Memory)-based model was developed to evaluate the impact of incorporating synthetic data on forecasting accuracy, particularly focusing on general trends and extreme market events. The findings reveal that models trained on a mix of synthetic and real data significantly outperform those trained solely on historical data, improving predictive performance.

Keywords: quantum machine learning; synthetic data generation; financial time series prediction; generative adversarial networks (GANs)

check for updates

Citation: Orlandi, F.; Barbierato, E.; Gatti, A. Enhancing Financial Time Series Prediction with Quantum-Enhanced Synthetic Data Generation: A Case Study on the S&P 500 Using a Quantum Wasserstein Generative Adversarial Network Approach with a Gradient Penalty. *Electronics* **2024**, *13*, 2158. https://doi.org/10.3390/electronics13112158

Academic Editors: Hao Xue and Du Huynh

Received: 9 March 2024
Revised: 27 May 2024
Accepted: 29 May 2024
Published: 1 June 2024

1. Introduction

In recent years, the finance industry has witnessed a paradigm shift towards leveraging advanced computational techniques for modeling, forecasting, and analyzing financial markets. Among these techniques, synthetic data generation has emerged as a revolutionary approach, offering new perspectives for overcoming challenges inherent in financial data analysis, particularly due to the limitations of real-world data regarding availability, privacy concerns, and the representation of rare events.

This paper introduces a methodology for generating synthetic data about the S&P 500 index, utilizing Quantum Wasserstein Generative Adversarial Networks (QWGANs). This approach represents a confluence of quantum computing and advanced Machine Learning, aimed at capturing and replicating the intricate dynamics of financial markets. The S&P 500 index, a bellwether for the U.S. economy, presents a complex and dynamic system influenced by many factors, making its accurate modeling and prediction a daunting task. Traditional financial modeling techniques often grapple with capturing the full spectrum of market behaviors, especially when it comes to predicting extreme market movements or black swan events, which, though infrequent, have profound impacts on the market. The advent of quantum computing offers unprecedented computational capabilities, potentially enabling the modeling of complex systems more accurately than classical computing. In this context, the application of Quantum Wasserstein Generative

Adversarial Networks (QWGANs) for synthetic data generation marks a significant innovation. QWGANs leverage the principles of quantum mechanics to model the probability distributions of financial indices, such as the S&P 500, facilitating the generation of data that mirror their real-world complexities.

The primary aim of this work is to explore the effectiveness of QWGANs in generating synthetic data that accurately reflect the dynamics of the S&P 500 index. Furthermore, this paper expands its analysis to include the Brazilian stock market index (BOVESPA) to demonstrate the generalizability of the methodology employed.

By synthesizing data that include representations of extreme market conditions, this work intends to bridge the gap in datasets where such events are under-represented. Subsequently, the synthetic data generated by QWGANs will be utilized to train a Long Short-Term Memory (LSTM) model to assess the impact of this synthetic dataset on the accuracy of S&P 500 forecasts. This investigation will include a comparative analysis of the performance of models trained with various combinations of real and synthetic data, emphasizing the utility of synthetic data in enhancing predictive models.

The novel aspect of this study is its emphasis on forecasting extreme events within stock markets. Given the rarity yet significant impact of such events, traditional datasets often lack sufficient instances to train models effectively for their prediction. By employing QWGAN-generated data, which include simulated instances of extreme market behavior, this research seeks to improve the predictive models' ability to forecast such events accurately. In this sense, this study is poised to contribute significantly to the field of financial forecasting and risk management by demonstrating the potential of quantum-enhanced synthetic data in improving the accuracy and robustness of predictive models. Moreover, by focusing on the prediction of extreme market events, this research addresses a critical gap in current financial modeling practices, offering insights into more effective strategies for risk assessment and management in the face of market volatility.

This article is structured as follows. Section 2 reviews the related work. Section 3 provides the foundations of classic quantum computing and Quantum Machine Learning. Section 4 presents a case study and the corresponding model. A second case study (BOVESPA) is proposed to further corroborate the approach's generalization capabilities. Section 5 reviews the results of the experiments, and, finally, Section 6 concludes this work.

2. Related Work

Quantum Generative Adversarial Networks (QGANs) are at the forefront of combining quantum computing with AI, with the potential to outperform traditional GANs in computational efficiency and data handling.

Huang et al. [1] present a detailed study related to the use of the Hadamard test quantum algorithm for obtaining gradients required for training QGANs. This method, which is crucial for training efficiency, is implemented by exploiting a superconducting quantum processor including five frequency-tunable transmon qubits. The experiment pursued by the authors proves the feasibility of QGANs in practical quantum computing environments. Dallaire et al. [2] delve into the quantum cost function, which is a specific component in optimizing QGAN performance. By tuning this function, QGANs can be properly trained to produce desired outputs. In this sense, the authors' work highlights the importance of theoretical foundations in advancing QGAN technology.

However, this kind of model faces challenges, including the nascent stage of quantum computing technology and training stability issues similar to those in classical GANs. The integration of classical data into quantum processes demands sophisticated encoding methods, with QGANs requiring efficient algorithms for both data processing and model training. S. Chakrabarti et al.'s development of a quantum Wasserstein GAN (QWGAN) [3] demonstrates significant advancements in processing speed and data handling through innovative loss functions and quantum-specific methodologies. B.T. Kiani et al. [4] introduce a QWGAN model that utilizes the quantum Earth Mover's (EM's) distance, showcasing its effectiveness in overcoming optimization challenges and noise sensitivity.

This model's success in generating complex quantum states and its practical application in real-world scenarios underscore the quantum EM distance's potential in enhancing quantum learning processes.

On the anomaly detection front, Herr et al. [5] explore a quantum variational GAN, AnoGAN, using a hybrid quantum–classical approach to improve training stability and efficiently process classical data without conversion to quantum states. Their work suggests quantum AnoGAN's comparable performance to classical models, indicating the feasibility of quantum approaches in specific scenarios. Situ et al. [6] tackle the vanishing gradient problem in generating discrete data distributions with a QGAN, employing a hybrid architecture to efficiently produce complex data patterns. Their approach demonstrates the capability of quantum models to efficiently generate discrete data, leveraging quantum circuitry to enhance model performance.

Kalfon et al. [7] address anomaly detection in time series with a cQGAN model, introducing Successive Data Injection (SuDaI) encoding to optimize data representation within limited qubit resources. Their model, tested on various datasets, shows high accuracy in anomaly detection, enhanced by dynamic thresholding to manage noisy data effectively.

Lastly, in [8], Ostaszewski et al. study the functional relationship between quantum control pulses in ideal conditions and those affected by an unwanted drift, utilizing LSTM neural networks to model and correct for these disturbances efficiently. The methodology section is divided into two parts, i.e., (i) a model of the quantum system describing a model involving two interacting qubits, influenced by a control and a drift Hamiltonian, using the GKSL master equation, and (ii) an architecture detailing the use of LSTM networks, suitable for time series data like control pulses, to approximate the correction needed to counteract the effects of drift. The authors discuss the performance of the neural network in generating control pulses that counteract drift, stating that the trained LSTM networks are shown to effectively create control pulses with high fidelity. Furthermore, a notable result consisting of the network's ability to adapt to local variations in control pulses and maintain high fidelity under changing conditions is tested and confirmed. The authors conclude by remarking on the effectiveness of using LSTM neural networks for modeling and correcting quantum control schemes in noisy environments, suggesting that this approach is promising for the further development of quantum technologies.

The article delves into the domain of quantum mechanics, focusing on the correction of quantum control pulses affected by drift using LSTM neural networks. The authors' research is crucial for enhancing the precision and stability of quantum systems, offering new methodologies in an area traditionally governed by quantum physics and control theory. By employing LSTMs, the study presents a novel way to model and mitigate complex quantum behaviors, which could significantly influence future quantum technologies by enhancing system reliability and performance.

Conversely, the present article navigates the financial sector, introducing the application of Quantum Wasserstein Generative Adversarial Networks (QWGANs-GP) to generate synthetic data that enhance the prediction accuracy of financial time series, such as the S&P 500 index. This study not only represents an innovative blend of quantum computing and generative adversarial networks but also marks a significant step towards integrating quantum-enhanced methods into financial forecasting. By improving the accuracy of models, especially in predicting rare but impactful extreme market events, this research could substantially influence risk management and market prediction strategies, potentially leading to more robust financial systems.

Both articles showcase the potential of quantum computing to transcend traditional boundaries and solve complex problems in diverse fields through hybrid quantum–classical systems and neural network applications. While [8] advances quantum computing in theoretical and practical realms within physics, this article explores its pragmatic use in economic forecasts, underscoring the broad applicability and transformative potential of quantum technologies.

3. Background

This section provides an overview of quantum computing applications (by summarizing the Shor and Grover algorithms and the concept of Variational Quantum Algorithms), Quantum Machine Learning (by describing the principles of Wasserstein GAN and the WGAN with Gradient Penalty), and, finally, the Quantum Wasserstein Generative Adversarial Network with Gradient Penalty.

3.1. An Overview about Quantum Computing

A quantum algorithm [9] constitutes a sequential methodology executable on a quantum computing platform, characterized by operations that leverage the unique principles of quantum mechanics, such as entanglement and superposition. These foundational quantum phenomena enable quantum algorithms to potentially execute certain computational tasks more efficiently than their classical counterparts.

Shor's algorithm [10] epitomizes the quantum computational advantage by efficiently decomposing an integer N into its prime factors. This quantum algorithm demonstrates a significant computational speedup over the best-known classical factoring algorithms. Specifically, Shor's algorithm operates within polynomial time complexity, denoted as $O(\log N)$, indicating that the required computational resources increase logarithmically with the input size N. In contrast, classical factoring algorithms exhibit an exponential time complexity, represented as $O(2^N)$, underscoring the profound efficiency of quantum algorithms in solving certain computationally intensive problems. Shor's algorithm encompasses two integral segments: quantum mechanics-based operations and classical pre-processing tasks. Initially, the classical segment reduces the integer factorization problem to the determination of a function's periodicity, a task achievable with conventional computing resources. Subsequently, the quantum segment undertakes the actual factorization by accurately identifying the period of the function, a process known as period finding, facilitated by the quantum computational environment. The practical deployment of Shor's algorithm encounters constraints related to the quantum computing infrastructure, specifically that the algorithm's demand for qubits directly correlates with the magnitude of the integer to be factorized. Presently, the nascent stage of quantum computing technology, characterized by a limited qubit count and elevated error rates, presents substantial challenges in applying Shor's algorithm for decrypting encryption schemes based on large integers, such as RSA keys [11]. RSA, a pivotal public-key cryptosystem, utilizes a pair of keys for encryption and decryption: a public key derived from two large prime numbers for encryption, and a private key, kept confidential, for decryption. Only the possession of the prime numbers, constituting the private key, enables message decryption.

Quantum Error Correction (QEC) emerges as a solution to counteract decoherence by employing additional qubits to encode quantum information redundantly, thereby enabling error detection and correction. Despite its potential, the high resource demands of QEC pose significant challenges to the practical deployment of quantum computers in the near term [12]. The efficacy of a quantum computer is not solely dependent on the quantity of qubits but also on their quality. Key metrics such as energy relaxation and coherence time gauge the stability and operational integrity of qubits. However, the prevailing noise levels in quantum systems significantly exceed signal strengths, rendering the current QEC approaches insufficient for achieving scalable quantum computing.

Variational Quantum Algorithms (VQAs) represent a forefront approach within the NISQ paradigm, employing a methodology akin to neural network training. These algorithms seek to optimize quantum circuits' parameters to either minimize or maximize a cost function related to a specific model, thus tackling optimization problems through a confluence of quantum and classical computational efforts. The quantum segment of VQAs, the Variational Quantum Circuit, processes input data to produce a measurable output influenced by adjustable parameters, with the quantum component responsible for executing the cost function and measurements, and the classical segment handling parameter optimization. VQAs draw inspiration from the Variational Principle, a methodological framework aimed at deriving approximate solutions to analytically intractable problems, such as obtaining the ground state energy of a quantum system via the Schrödinger equation. This principle involves hypothesizing a wave function, $|\psi(x)\rangle$, contingent on variational parameters, x, which are iteratively adjusted to minimize the energy estimate of the wave function. According to the Variational Theorem, the energy estimate provided by any trial wave function serves as an upper limit to the actual ground state energy, E_0, thus directing the optimization process towards the minimal energy configuration. VQAs are adaptable to specific problems yet adhere to a structured framework, incorporating distinct phases for their operation as illustrated in the architecture diagram of a Variational Quantum Algorithm (see Figure 1).

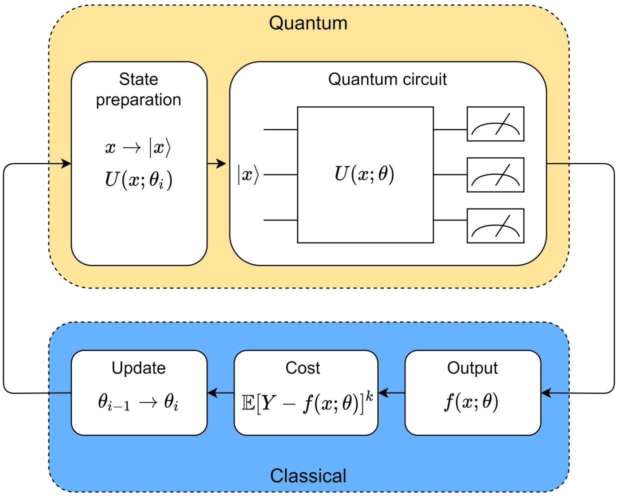

Figure 1. Architecture of a Variational Quantum Algorithm.

3.2. Quantum Machine Learning

Quantum Machine Learning (QML) extends the principles of Machine Learning into the quantum domain, aiming to harness quantum computational speedups for data processing and algorithmic learning. QML offers novel approaches to processing and learning from data in ways that classical computing cannot achieve alone. This synergy promises significant advancements in computational efficiency and algorithmic complexity, particularly in the era of Noisy Intermediate-Scale Quantum (NISQ) devices and beyond. Generative models constitute a paradigm that can synthesize novel data instances, contrasting with discriminative models which discern between differing data instances. Generative models encapsulate the underlying data distribution, facilitating the estimation of the likelihood of specific instances. Conversely, discriminative models predict the probability of labels given the data instances. Generative modeling autonomously identifies and learns patterns within input data, enabling the synthesis of new examples that plausibly align with the original dataset.

A notable instance of generative models is the Generative Adversarial Network (GAN), conceptualized by Goodfellow et al. [13]. A GAN comprises two core components:

- Generator (G), which synthesizes new data instances from a randomly selected vector following a Gaussian distribution.
- Discriminator (D), a binary classifier tasked with distinguishing between genuine data (real) and synthetic data (fake) produced by the generator.

These two networks engage in a competitive dynamic, where the generator seeks to deceive the discriminator, which in turn adapts to more accurately classify new synthetic data. This iterative process enhances the generator's ability to produce data increasingly indistinguishable from actual data. The discriminator's output, typically derived from a sigmoid function, quantifies the probability of data being real, with $D(x)$ representing this probability. The discriminator aims to minimize a cross-entropy loss function, while the generator endeavors to maximize it, indicating a min–max game between the two networks. The process is visualized in Figure 2, illustrating the iterative training cycle that refines both networks.

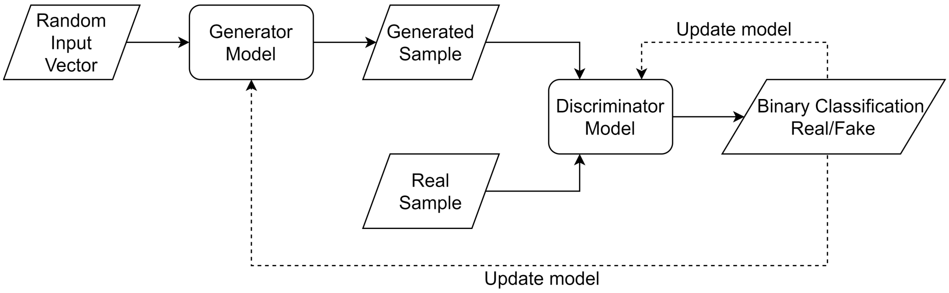

Figure 2. GAN's architecture.

The Wasserstein GAN (WGAN) [14,15] introduces a novel cost function grounded in the Wasserstein metric, also known as the Earth Mover's distance (EMD), offering a measure of the cost to transform one data distribution into another. The WGAN framework aims to minimize this distance, facilitating a more stable and convergent training process compared to traditional GANs. This approach requires the discriminator, here referred to as the Critic, to approximate a 1-Lipschitz function, guiding the generator to produce data that minimize the Wasserstein distance between the synthetic and real data distributions. An iteration of the WGAN, the WGAN with Gradient Penalty (WGAN-GP), incorporates a gradient penalty into the Critic's loss function to enforce the Lipschitz constraint more effectively. This adjustment aims to stabilize the training process and mitigate common issues such as mode collapse and vanishing gradients, marking a significant advancement in the GAN methodology. Quantum GANs (QGANs) aim to replicate the probabilistic distribution of original data using quantum computational advantages. These models leverage quantum algorithms to potentially surpass classical GANs in learning efficiency and data generation capabilities. The QGAN architecture can vary, including both fully quantum and hybrid quantum–classical configurations, with the latter often employing a quantum generator alongside a classical discriminator to optimize performance and feasibility in current quantum computational paradigms.

QGANs embody a pioneering amalgamation of artificial intelligence and quantum computing technologies. This emergent methodology, which has yet to be fully validated, possesses the potential to eclipse the performance benchmarks established by traditional classical GANs. According to Ngo et al. [16], QGANs are poised to surpass classical GAN models in the following key areas:

- Computational Power: The principle of quantum supremacy suggests that quantum models can execute certain computations exponentially faster than their classical counterparts. Within QGAN frameworks, this attribute frequently translates to a reduced parameter requirement to achieve or exceed the performance levels of classical GANs.
- Data Handling Capabilities: Quantum computing exhibits a unique proficiency in processing, manipulating, and generating particular data types, offering a distinct advantage in scenarios where classical computational approaches encounter limitations.

In particular scenarios, the hybrid quantum–classical computation paradigm, integrating a quantum generator with a classical discriminator, demonstrates superior performance compared to entirely classical GAN models. This advantage primarily stems from the accelerated and simplified convergence during the training phase. However, QGANs also encounter specific challenges. Notably, the full potential of quantum computation remains untapped, with current applications still navigating the NISQ era, affecting all quantum algorithms. Furthermore, akin to classical GANs, QGANs are susceptible to training instabilities, potentially complicating the achievement and maintenance of convergence.

Additionally, the risk of model collapse, where the generator produces a limited or identical output set, is present in both quantum and classical models [17].

A significant hurdle in QGAN development involves the integration of classical data with quantum operations. When both the generator and discriminator are quantum-based, classical data must be encoded into quantum states for input into the discriminator. Conversely, if only the generator is quantum, the generated data must be converted back into a classical format for further analysis or application. This necessitates a careful consideration of the interplay between classical and quantum data processing in QGANs.

Building on the foundational concepts of GANs, Wasserstein GANs (WGANs), and the advancements introduced by the WGAN with Gradient Penalty (WGAN-GP), the Quantum Wasserstein Generative Adversarial Network with Gradient Penalty (QWGAN-GP, [18]) represents a pivotal evolution in the domain of QML. This model integrates the quantum computing framework with the WGAN-GP architecture to address and potentially surpass some of the limitations faced by classical GANs and QGANs, particularly focusing on training stability and the quality of generated data. The QWGAN-GP model leverages quantum circuits to construct both the generator and the discriminator (or Critic, in the context of WGANs), exploiting quantum mechanical properties to enhance the generative modeling process. The quantum generator creates quantum states that approximate the distribution of a given quantum dataset, while the quantum discriminator evaluates the authenticity of these generated states against real quantum data samples. The inclusion of the gradient penalty term in the loss function, as inspired by WGAN-GP, aids in enforcing the Lipschitz constraint more effectively, which is crucial for the training stability and convergence of the model. By incorporating the gradient penalty, QWGAN-GP mitigates common issues such as mode collapse and the vanishing gradient problem, which are prevalent in classical GANs and some QGAN models. Moreover, quantum systems inherently provide an exponential increase in state space with a linear increase in qubits, allowing QWGAN-GP models to potentially represent and process complex data distributions more efficiently than their classical counterparts. QWGAN-GP models hold promise for applications in quantum chemistry, material science, and quantum cryptography, where they can be used to generate and analyze quantum states that are difficult to simulate classically. Despite their potential, QWGAN-GP models face challenges related to the current limitations of quantum hardware, such as coherence times, qubit connectivity, and error rates. These hardware constraints necessitate efficient quantum circuit designs and error mitigation techniques to realize the practical deployment of QWGAN-GP models. Furthermore, the complexity of training quantum models and the need for significant classical computational resources for simulation and optimization highlight the importance of hybrid quantum–classical approaches in the near term.

4. Case Study and Problem Formulation

This section details a case study utilizing a QWGAN-GP model (the code used for the experiments, together with the instructions to execute them, can be found on the Github repository https://github.com/EBarbierato/Enhancing-Financial-Time-Series-Prediction-) for synthesizing data corresponding to the S&P 500 index. It implements the model's quantum generator and classical Critic architectures (the architecture used to perform the experiments consists of qBraid Lab, a comprehensive cloud platform tailored for quantum computing, equipped with a custom notebook environment that integrates with tools like GitHub and VSCode. It provides access to numerous quantum computing resources, including over 20 quantum computers and simulators via the qBraid SDK. The SDK allows for easy programming across different quantum frameworks and seamless execution on quantum hardware. For example, users can create quantum circuits using one framework, like PennyLane, then transpile these using another, such as Qiskit, and execute them on a third platform like Amazon Braket, showcasing the SDK's versatile application capabilities). Moreover, it explores various S&P 500 predictions using an LSTM model with different synthetic datasets to ascertain the optimal data combination for prediction accuracy. The dataset, derived from the “yfinance” Python library, spans 1 January 2000 to 31 December 2008, a period chosen for encompassing the Great Financial Crisis, characterized by significant market volatility and the occurrence of extreme events, thus reflecting the broader U.S. economic landscape (see Figure 3 with daily resolution).

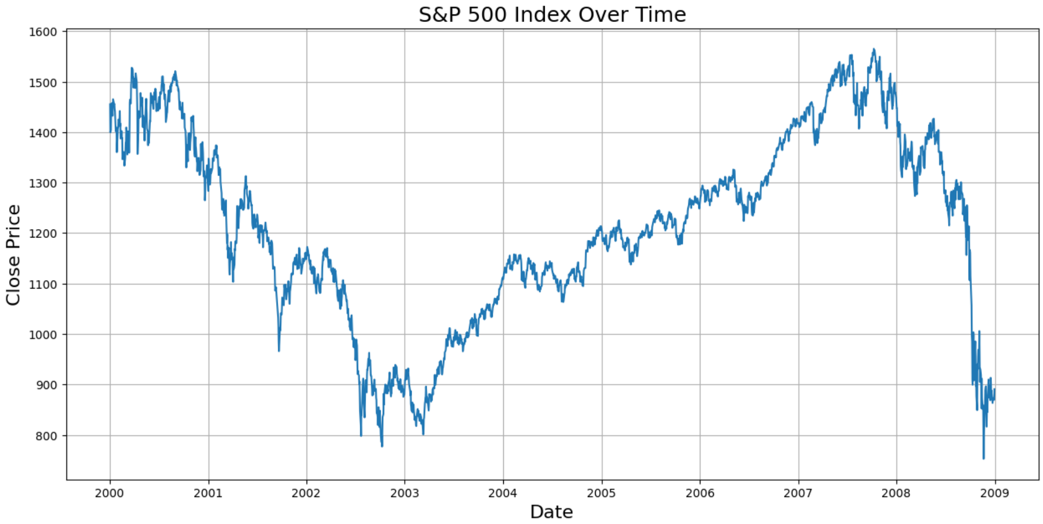

Figure 3. S&P 500 Close price values over time.

The model’s primary contribution lies in introducing an alternative Parametric Quantum Circuit (PQC) designed to be computationally less complex while retaining efficiency. It treats data points as a series of logarithmic returns, a choice reflecting the inadequacy of solely studying index prices for comparing performance across different periods. Logarithmic returns are favored in financial time series analysis due to their properties, including value compression and additivity over time, which facilitates the summation of returns across periods to calculate the total return for the entire time frame, as highlighted in econophysics studies.

4.1. Data Generation

The logarithmic returns can be calculated as per Equation (1), which takes the logarithm of the Close price at time t and subtracts from it the logarithm of the price at time $t-1$.

$$r_t = \ln(p_t) - \ln(p_{t-1}) \tag{1}$$

Figure 4 shows the evolution of the logarithmic returns of the S&P 500 over time.

Figure 5 depicts the Probability Density Function (PDF) of the logarithmic returns for the S&P 500 index, indicating a divergence from a normal distribution. This divergence is marked by an elevated peak density and tails that are more persistent compared to the expected rapid decay in a normal distribution. To rectify this discrepancy, normalization was applied to adjust the logarithmic returns to a mean of zero and a standard deviation of one, standardizing all Log Returns to a consistent scale. The specific normalization formula utilized is detailed in Equation (2):

$$r_{t_{\text{NORM}}} = \frac{r_t - \mu}{\sigma} \tag{2}$$

To address the pronounced heavy tails in the distribution, diverging from normality, a technique utilizing the inverse Lambert-W transform on normalized logarithmic returns was employed, aiming to modify the distribution towards Gaussian characteristics. Lambert’s W function, the inverse function of $z = ue^u$, facilitates a transformation suitable for datasets with pronounced heavy tails, such as the normalized dataset V. This method, as delineated by Goerg et al. [19], enables the reshaping of the distribution to more closely align with a Gaussian distribution, as detailed in Equation (3):

$$W_\delta(v) = \text{sgn}(v)\sqrt{\frac{W(\delta v^2)}{\delta}} \tag{3}$$

where v is a value in a dataset, $\text{sgn}(v)$ is the sign of v, δ is a tunable parameter, and W is the Lambert W function.

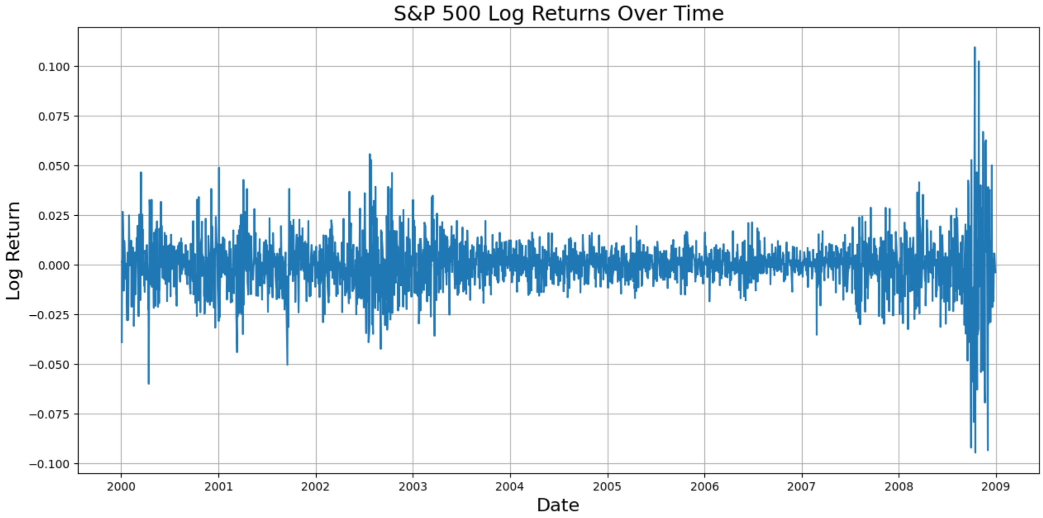

Figure 4. S&P 500 Log Return values over time.

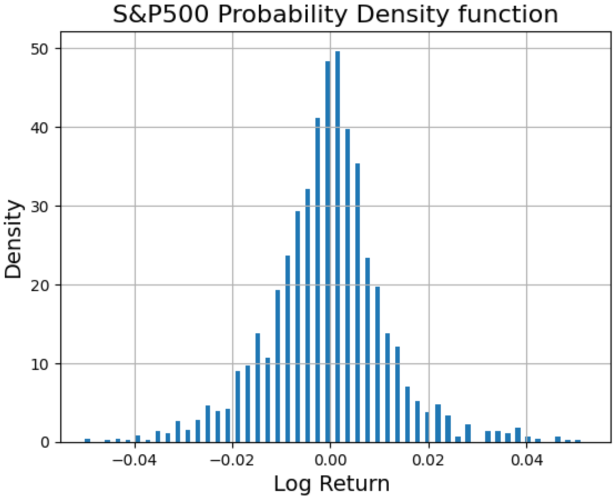

Figure 5. S&P 500 Probability Density Function.

The rolling window technique generates a series of overlapping subsequences from the original time series data, capturing temporal patterns and dependencies. It relies on two parameters: the subsequence length (m) and the stride (s), which is the gap between the starts of subsequences. Typically, the window length (m) is set longer than the stride (s) to ensure sufficient overlap. For this analysis, m is set to 10 and s to 2, optimizing the balance between overlap and computational efficiency.

The model to generate synthetic data for the S&P 500 index was trained on 2000 epochs and the size of the batch, so the number of training samples used in one iteration of the model training process to update the model's parameters is equal to 20. The quantum generator plays a crucial role in attempting to produce a data distribution that closely approximates the true underlying distribution. In the model under discussion, the quantum circuit is composed of five qubits and is structured with three layers. It incorporates single-qubit rotation gates to manipulate individual qubit states, entangles layers to create quantum correlations between qubits, and re-uploads layers. This multi-layered approach is designed to enhance the generator's ability to capture the complexities of the S&P 500 data distribution.

In the quantum circuit for generating synthetic S&P 500 data, measurements are carried out on each qubit using both the Pauli-X and Pauli-Z operators. Specifically, the setup involves two distinct types of measurements: one on the X basis (Pauli-X) and one on the Z basis (Pauli-Z). Consequently, each qubit encounters measurements with both operators, allowing for the extraction of information about the qubit states on both the corresponding X and Z bases. This methodology is advantageous as the X operator assesses the quantum state in the superposition basis, distinguishing between the $|+\rangle$ and $|-\rangle$ states, which represent the qubit being in an equal superposition of being in state $|0\rangle$ or $|1\rangle$, but with different relative phases. On the other hand, the Z operator measures the quantum state in the computational basis, which corresponds to the $|0\rangle$ and $|1\rangle$ states. Combining these measurements allows for the understanding of the qubit's behavior and properties.

The encoding layer is responsible for setting the initial state of the qubits in the quantum generator. In this case, the quantum generator circuit is initialized with a different set of values by using random noise. Specifically, each value is between 0 and 2π and each qubit receives a unique random angle for its initialization. Finally, the R_x gate is applied to prepare the initial quantum state of each qubit, with the parameter angle given by the random values cited before. This method helps the generator explore a wide range of possible outputs.

The PQC was created with the following method:

- Hadamard Gate: A Hadamard Gate was applied to each qubit at the beginning of the circuit, transforming the initial state of each qubit, that is, $|0\rangle$, into an equal superposition of both basis states, $|0\rangle$ and $|1\rangle$.
- Rotation Layers: The decision to exclusively utilize the R_x and R_y gates within the quantum generator circuit was deliberate and was based on their impact on the qubit amplitudes. In contrast, the R_z gate primarily influences the phase of the qubit. Consequently, focusing on the R_x and R_y gates allows for precise control over the qubit amplitudes, which is particularly relevant for shaping and manipulating the quantum state to achieve the desired outcomes in the generative process.
- Entangling layers: In the model, the entangling layers were constructed using CNOT gates, which are non-parametrized gates. These CNOT gates were applied to every pair of qubits within the quantum circuit. The purpose of using CNOT gates in this manner is to establish entanglement between all pairs of qubits in the quantum circuit.

The presence of three layers in the model results in the repetition of both rotation layers and entangling layers three times.

Figure 6 shows part of the PQC used in the quantum generator.

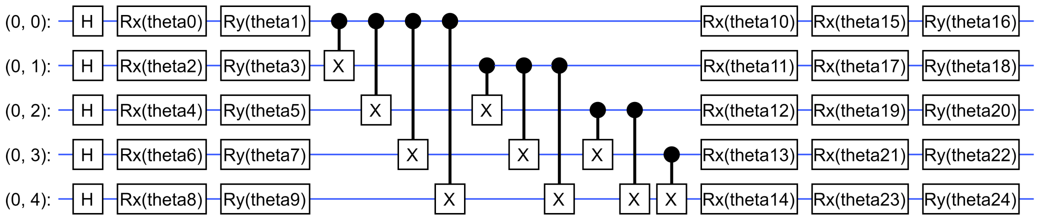

Figure 6. Encoding and single-layer representation of the quantum generator's PQC.

The Critic model is a convolutional neural network (CNN) designed to process sequential data. It takes the data generated by the quantum generator and learns to distinguish between real (original) data and fake (generated) data.

The Critic model comprises three convolutional layers with increasing numbers of filters: 64, 128, and 128. This incremental increase in the number of filters enhances the network's capacity to extract intricate features from the input data, resulting in more detailed feature maps as output. In particular, the first convolutional layer takes into account the shape of the input data, which are in the form of $(10, 1)$. The number 10 corresponds to the length of the subsequences generated using a rolling window approach. This consideration ensures that the network processes the input data in a manner that respects their temporal structure.

After each convolutional layer, a LeakyReLU activation function is applied with a factor of $\alpha = 0.1$.

$$\text{Leaky ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{else.} \end{cases} \tag{4}$$

The LeakyReLU function described by Equation (4) helps to prevent the dying ReLU problem where neurons stop learning and allows the model to better capture complex relationships, due to its nonlinearity [20].

After the three convolutional layers, the data go through flattening, which is a common practice when transitioning from convolutional layers to fully connected layers. This flattening reshapes the data into a one-dimensional vector, making them suitable for feeding into a dense layer. Following the flattening step, a dense layer with 32 neurons is employed, and the same LeakyReLU activation function is applied as before. Dense layers are well suited for capturing complex data relationships. To prevent overfitting, a dropout layer is introduced after the dense layer, where 20% of the neurons are randomly omitted during the forward pass, helping improve the model's generalization performance. Finally, the output layer consists of a single neuron, which produces a single scalar value as the model's output. In general, there is no bound for the output value and higher values indicate that the input is evaluated as real data, while lower values indicate the contrary. The unbounded nature of the Critic's output is related to the use of the Wasserstein distance as a measure.

The cost function employed is the Wasserstein distance. To improve the training stability of the model, it used a penalty term, enforcing the Lipschitz constraint.

Figure 7 depicts the two time series, with the original one in blue, while the one regarding the data generated synthetically is in orange.

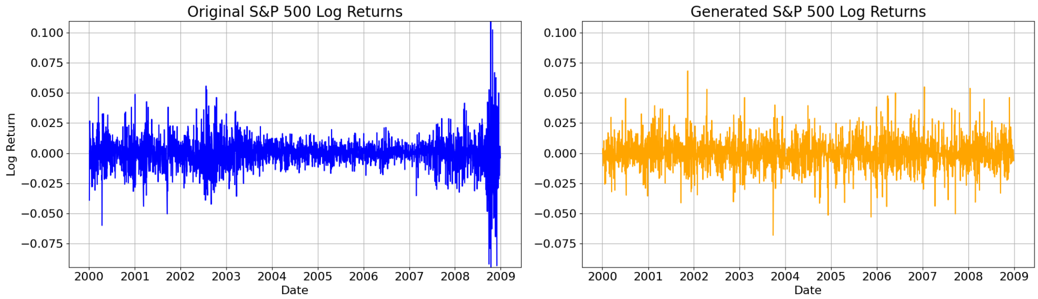

Figure 7. Plot of the time series related to the original Log Returns and the generated Log Returns.

The results achieved with the previously mentioned model are promising. To assess the characteristics of the two distributions, several metrics were employed for comparison. One of the initial comparisons involves the use of a quantile–quantile (QQplot) plot to visually assess the similarity or divergence between the two distributions.

In Figure 8, it is observed that both distributions, the original data distribution and the synthetic data distribution generated by the model, deviate from normality. However, there are some differences in their characteristics. The distribution of the original data exhibits greater skewness, indicating that it deviates more significantly from a normal distribution. Interestingly, in the central region, encompassing the quantiles between -2 and 2, both distributions closely follow a normal distribution. This suggests that a substantial proportion of observations from each distribution are similar and exhibit behavior consistent with a normal distribution within this range.

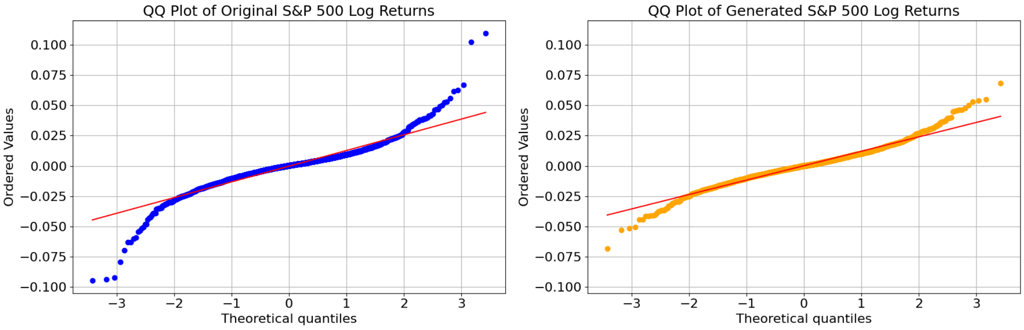

Figure 8. QQplot comparison between the original Log Returns and the generated Log Returns.

Next, the PDFs and CDFs of both distributions are compared.

Figure 9 shows that in the central part, where most of the changes are included, the two distributions are pretty similar. In the tails, the original data present more extreme events, with higher values, compared to the generated one, as the QQplot also highlights. However, this representation does not perfectly reflect the two distributions because the bins are a discretization of the Log Return values. Indeed, in Figure 10, which represents the continuous PDF, the two probability distributions nearly overlap each other. As evidence of this high similarity, the Wasserstein distance, defined as the minimum amount of "work" required to transform one distribution into the other, between the two distributions was calculated, and it was found to be 0.0008608819538412148, which is nearly close to zero.

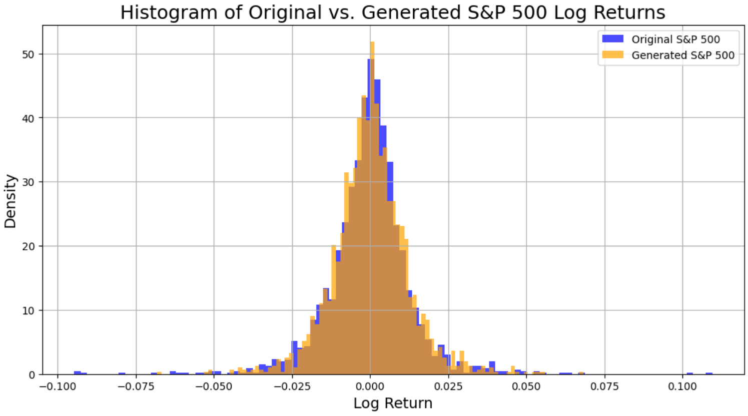

Figure 9. Histogram comparison between the original Log Returns and the generated Log Returns.

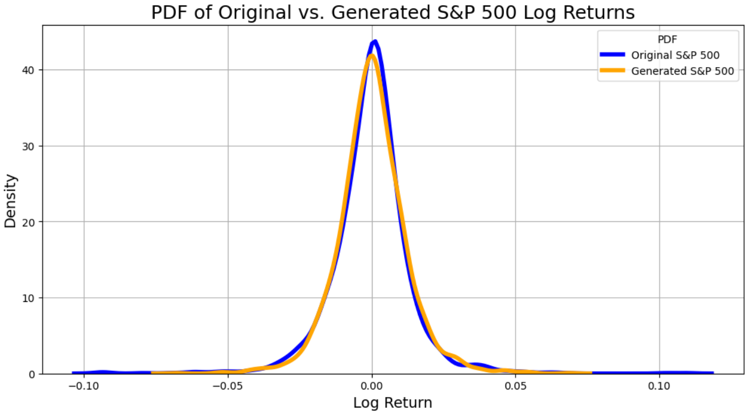

Figure 10. PDF comparison between the original Log Returns and the generated Log Returns.

The CDF representation in Figure 11 shows that the two distributions are very similar; this graph highlights the probability that a Log Return picked at random is less than or equal to that value. The orange line closely follows and overlaps the blue line, meaning that the two distributions have the same probability.

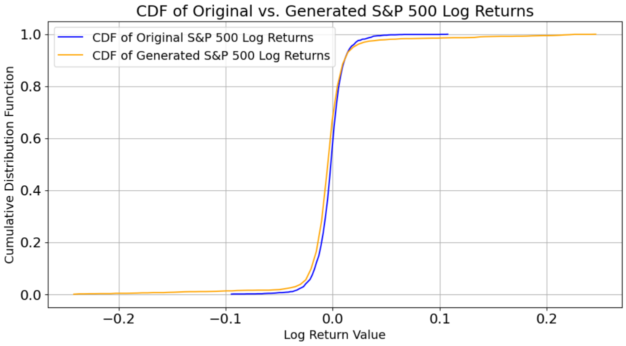

Figure 11. CDF comparison between the original Log Returns and the generated Log Returns.

In addition, the two autocorrelation functions (ACFs) of the two time series are depicted below in Figure 12.

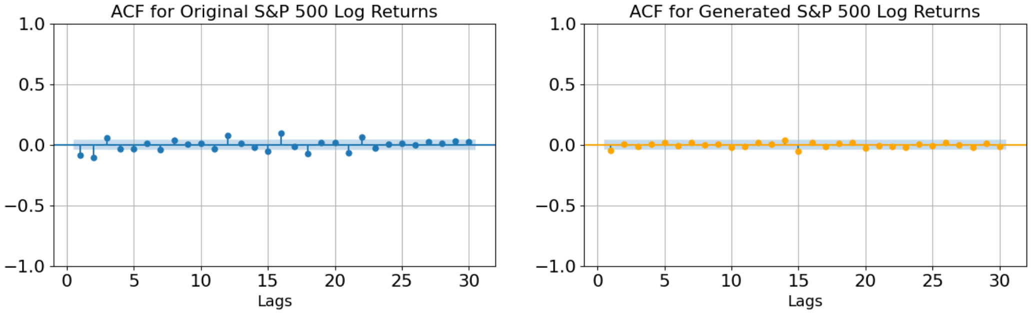

Figure 12. ACF comparison between the original Log Returns and the generated Log Returns.

Both ACF plots show that the autocorrelations across various lags are generally low, tending towards zero and without statistical significance, which is consistent with the properties of efficient financial markets where each Log Return is mostly independent of the past. This leads to uncorrelated behavioral patterns due to the rationality of investors, who make well-founded decisions based on all available information.

Moreover, entropy is also taken under consideration. It is a measure of the uncertainty in the probability distribution of the Log Returns. A higher value of the entropy indicates a more unpredictable distribution. In this case, both entropies are very close to each other. Indeed, for the original logarithmic returns, the entropy is 3.1528, while that for the generated data is 3.1289. This indicates that both data points have a similar degree of randomness.

In addition, the Dynamic Time Warping (DTW) distance was computed between the two time series. Despite having similar distributions, it is important to note that the two series are not necessarily identical in their actual values, which can result in warping or stretching along the time axis. The DTW distance measurement is valuable in this context because it calculates the degree of similarity between the two time series while accommodating for these nonlinear variations and potential temporal mismatches.

Figure 13 illustrates the DTW between the original series, represented on the y-axis, and the generated series, depicted on the x-axis. The red line in the graph represents the optimal warping path between the two series, with a window length set to 500. The warping path represents how the two series are aligned against each other in time. It determines how each point in one time series corresponds to one or more points in the other series.

A perfect warping path is equal to a straight diagonal line. It is evident from the plot that there is a noticeable time mismatch between the two series. This discrepancy suggests that the generated series represents a different plausible scenario compared to the original series. The DTW distance between the two series is computed to be 1.954. This relatively small value indicates that, despite the warping and time misalignment, the two series exhibit a significant degree of similarity. This similarity is reflected in the relatively low DTW distance, suggesting that the generated series captures important aspects of the original data and represents a plausible alternative scenario, even with temporal variations.

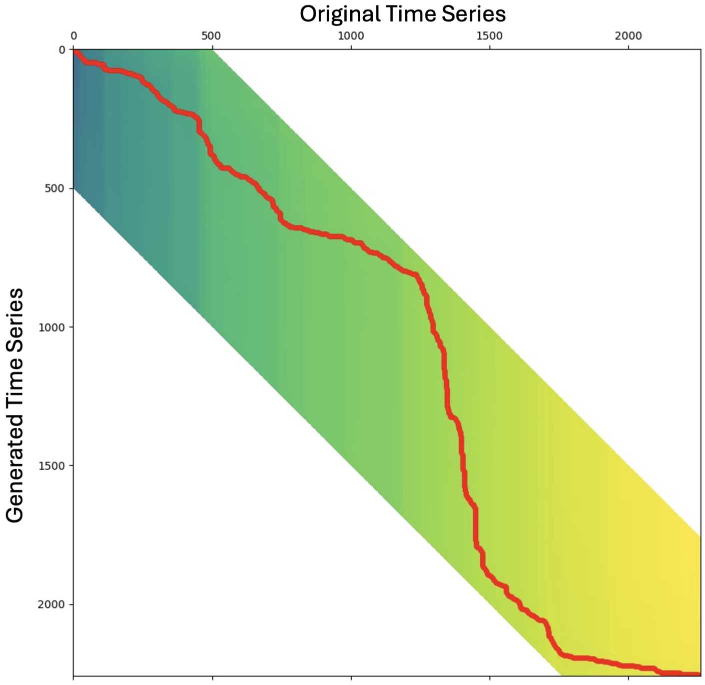

Figure 13. DTW distance between the original Log Returns and the generated Log Returns.

To conclude the comparison analysis, a scatter plot was generated to illustrate the two time series in a two-dimensional format. On the horizontal x-axis, the values of Log Returns at time t are represented, while on the vertical y-axis, the values of Log Returns at the previous time step, denoted as $t-1$, are depicted. This plot can be useful for detecting complex patterns that cannot be visualized with the one-dimensional representation.

Figure 14 shows that both real and generated data points have a cluster around the center, where the density of points is also higher, which suggests that the synthetic data capture the general behavior of the real data well. Data points do not show any clear pattern; it seems to be that they are randomly dispersed, confirming that there are no elements of autocorrelation. However, real data show more spreads with a higher number of outliers in the graph. This indicates a higher presence of extreme events in the real data than in the generated data. The presence of a pronounced concentration of data points towards the center, along with a significant occurrence of extreme events, implies the existence of a volatility cluster, particularly in the real data. This phenomenon suggests that large returns tend to follow large returns, and small returns tend to follow small returns, indicating a clustering effect in volatility. However, it is worth noting that the synthetic data exhibit fewer extreme values, and as a result, they may not entirely capture this specific aspect of the original data's behavior.

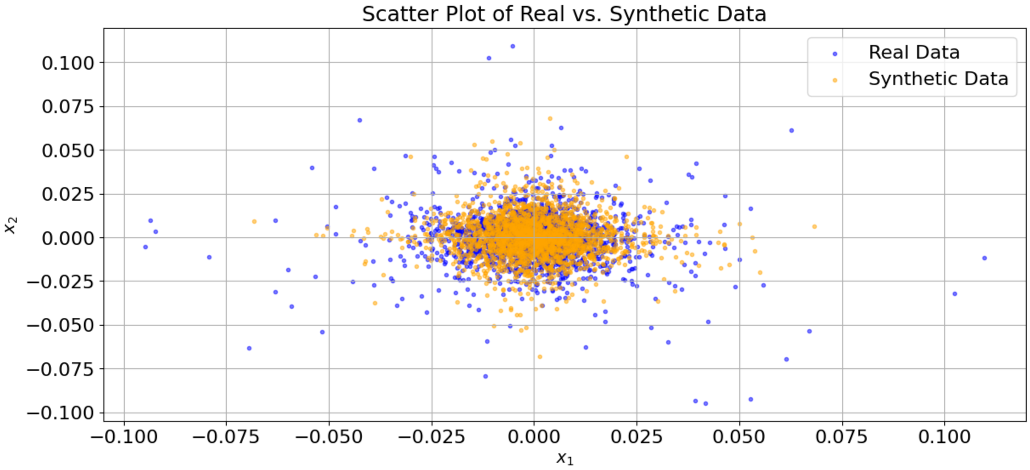

Figure 14. Lagged value comparison between the original Log Returns and the generated Log Returns.

4.2. Prediction Models

This subsection delves into the practical applications of the data generated through the QWGAN-GP model. Specifically, it explores various types of predictions, through an LSTM model which leverages the synthetic data in two primary ways. Firstly, this is accomplished as a feature incorporated into the model and secondly as a means to enhance extreme events within the original time series.

The assessment of the data generated involve an evaluation of the modle's effectiveness in two main aspects. Initially, its performance is gauged in making general predictions on the S&P 500 data. Then, a specific emphasis is placed on its ability to predict and handle extreme events within the dataset.

The historical time series data for both the original and synthetic Log Returns are combined in two distinct manners. Subsequently, these two combinations are subjected to analysis and compared with a benchmark model, which involves solely using the original series.

The first approach to using synthetic data involves incorporating the generated series as a feature within the model. This decision stems from the observation that the generated series closely approximates the distribution of the original data. By including the synthetic series as a model feature, it introduces an alternative scenario for the evolution of the S&P 500 series. This approach allows for exploring potential variations in the data's behavior, which is especially useful for the ability to generalize well on the test set.

The second approach involves enriching the original time series by incorporating extreme events from the synthetic series. Specifically, 82 extreme data points from the generated time series are added to the original time series. This method is particularly valuable for increasing the model's capability to predict extreme events when applied to the test set. With this method, the model gains greater exposure to extreme event scenarios, potentially improving its ability to anticipate and handle such events during predictions. In particular, an extreme event (EV) is defined as per Equation (5), where both μ and σ are calculated on the original logarithmic returns [21]:

$$EV = \mu \pm 2\sigma \tag{5}$$

Subsequently, a normalization step is performed, which involves scaling each feature within the dataset to a range between 0 and 1, ensuring that the data are within a consistent and standardized range, facilitating model training. Additionally, a transformation function is applied to make the data suitable for use with the LSTM network. In particular, sequences of 10 data points are created, and these sequences are used to predict the subsequent points in the time series. Each prediction is evaluated with the actual value of the original time series, like a supervised ML problem. This approach allows the model to learn from patterns and dependencies within these sequential data points.

All the above transformations were applied to each different prediction method.

The LSTM model is designed for a sequence prediction problem to predict the next value in a sequence based on previous observations. In this problem, it used a sequential model that arranges layers linearly. The same construction of the model was applied to each prediction model to highlight the comparative performance of each one.

The input layer consists of an LSTM layer with 50 neurons, which feed the input data with the shape of $(10, 1)$. The number 10 depends on the number of data points present in each sequence, while the 1 depends on the number of features used. The only exception was made for the feature model, which requires, for its nature, the addition of a second feature, meaning that this parameter was set equal to two. Another LSTM layer with 100 neurons was applied. After each LSTM layer, a dropout layer was applied with a rate of 0.2 to prevent overfitting. Finally, a dense layer with a single unit was added as the output layer, producing a single final value. The model was compiled with the Adam optimizer and the mean squared error (MSE) as a loss function. In conclusion, each training was performed for 100 epochs, updating the weights of the parameters after 32 samples.

5. Evaluation of the Results

The different models were evaluated by using the root mean squared error (RMSE) and the Mean Absolute Error (MAE) defined in Equations (6) and (7), respectively:

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j| \tag{6}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2} \tag{7}$$

where $\hat{y}_j$ is the predicted data point at time j, y_j is the actual data point at time j, and n is the total number of observations.

The results of the general S&P 500 predictions are presented in Table 1. These predictions refer to the testing set, which is composed of 23 data points.

Table 1. General prediction results on the testing set of the S&P 500 with different methods.

Data	RMSE	MAE
Original	0.00644	0.00558
Feature	0.00411	0.00338
Oversampling	0.00615	0.00502

As indicated in Table 1, the most favorable outcome was achieved using the Feature method, which appears to provide the best results. In contrast, employing solely the original time series data and the Oversampling method does not seem to yield substantial differences in performance.

It appears that the addition of extreme events to the training set did not result in a significant improvement in the model's performance. One potential explanation for this outcome could be related to the relatively small number of extreme events added. Specifically, incorporating 82 extreme events from a historical series comprising 2260 data points may not have provided a substantial impact on the overall training process.

In contrast, the Feature method stands out as a significantly improved approach compared to the benchmark. Indeed, it demonstrates an impressive increase in performance, achieving a 36.2% improvement in the RSME and 39.4% in the MAE over the original model. This notable performance improvement can be attributed to the inclusion of the synthetically generated data, which share the same probability distribution as the original model. By incorporating generated data as a feature, the model is effectively trained on an additional potential scenario of the S&P 500. As demonstrated by the results, this approach leads to an enhancement in the model's performance. It allows the model to consider and learn from this supplementary scenario, contributing to its improved predictive accuracy.

Table 2 presents the results of the model in predicting extreme events, as discussed earlier. The calculations were conducted on the training set, primarily due to the rarity of extreme events in the test set. This approach ensures a more significant presence of these specific events in the time series, facilitating the evaluation of the model's ability to predict and respond to extreme market occurrences.

Table 2. Extreme-event prediction results on the training set of the S&P 500 with different methods.

Data	RMSE	MAE
Original	0.04447	0.04134
Feature	0.02971	0.02764
Oversampling	0.04589	0.04355

The results align with the observations made in the general prediction analysis. Once again, the Feature method stands out as the best model, showcasing a significant increase in performance compared to the others. Specifically, both the RSME and MAE demonstrate notable improvements, with a substantial increase of 33.1%. The observed improvement can be attributed to the same principle as explained previously in the general prediction analysis of the S&P 500. Specifically, exposing the model to additional potential scenarios, as facilitated by the Feature method, has the potential to increase its performance.

On the contrary, the Oversampling method is even worse compared to the benchmark model. Training the model on a limited number of additional extreme events may not yield significant benefits, leading to higher RMSE and MAE values due to the increased presence of extreme events within the series.

Additional Case Study

In addition to conducting a detailed examination of the primary case study, this research extended its analysis to explore the Brazilian stock market index (BOVESPA). The motivation for this expansion stems from the unique characteristics of emerging markets, which differ significantly from established markets, exemplified by the S&P 500.

The time frame selected for this analysis was consistent with that of the previous case. Nonetheless, several entries labeled as "NaN" were encountered within the dataset. To ensure both continuity and accuracy, these missing values were rectified by substituting them with a four-day moving average. This approach was employed to preserve the integrity of the dataset and minimize the influence of missing data on further analyses.

The methodology retained the same model architecture used previously, applicable to both the generation of synthetic data and the implementation of the Long Short-Term Memory (LSTM) model.

Table 3 presents the general results obtained.

Table 3. General prediction results on the test set of the BOVESPA index with different methods.

Data	RMSE	MAE
Original	0.0177	0.0137
Feature	0.0092	0.0065
Oversampling	0.0241	0.0188

The results obtained for the general predictions are coherent with those observed for the S&P 500 index. This convergence underscores the robustness and versatility of the model across different market contexts for general predictions. Table 4 presents the results of the model in predicting extreme events.

Table 4. Extreme-event prediction results on the training set of the BOVESPA index with different methods.

Data	RMSE	MAE
Original	0.0596	0.0564
Feature	0.0585	0.0552
Oversampling	0.0689	0.0670

The findings from the case study on the prediction of extreme events indicate an improvement, albeit slightly less pronounced than that observed in the S&P 500 scenario. This highlights the model's capacity to generalize across various market contexts.

6. Conclusions and Future Work

This work discusses a QWGAN-GP constructed using a quantum generator and a classical discriminator, creating a synthetic time series to replicate the statistical properties of the S&500. The results obtained from this model demonstrate its effectiveness in comparison to the statistical properties of the original series. Various metrics, including Wasserstein distance, DTW distance, entropy measures, and others, are calculated, and in each case, the generated data perform robustly.

Furthermore, to reinforce the strength of the generated time series, an LSTM prediction model is created. This model assesses the impact of the generated data on prediction performance, both for general purposes and with a specific focus on extreme events. Remarkably, in both prediction scenarios, a combination of synthetic and real data outperform the benchmark established by only the original series.

The encouraging results obtained in this research open the way for ongoing work, especially in the area of Quantum Machine Learning. Researchers are actively working to address the limitations of quantum computation, such as the constraint imposed by a limited number of qubits to guarantee the resilience of algorithms against high levels of noise. In this specific case study, there is the potential to incorporate a quantum discriminator instead of the classical one to increase computational speed and higher accuracy. In addition, if the NISQ era limitations are surpassed, there will be the possibility to use a greater number of qubits and layers, increasing the complexity of the Parametrized Quantum Circuit.

Regarding future work, two aspects need to be considered. Firstly, the ability of the QWGAN-GP to handle data from different periods depends significantly on its training. A model trained on a diverse range of market conditions is more likely to generalize well across different periods. If the training data lack representation of certain market dynamics, the model's effectiveness could decrease when faced with these unrepresented conditions in separate periods. To enhance the model's performance across various periods, adjustments to the model parameters or the incorporation of period-specific features may be necessary. This could involve optimizing quantum circuits or modifying the loss functions used during the training phase. Secondly, it would be beneficial to explore the behavior across varying timescales by segmenting the data into distinct periods—for instance, comparing datasets spanning one year with those extending over five years. This approach will help in understanding how trends and patterns evolve over different durations.

Author Contributions: Conceptualization, F.O. and E.B.; methodology, E.B.; software, F.O.; validation, F.O., E.B. and A.G.; formal analysis, F.O.; investigation, F.O.; resources, F.O.; data curation, F.O.; writing—original draft preparation, F.O.; writing—review and editing, E.B. and A.G.; supervision, E.B. All authors have read and agreed to the published version of the manuscript. All authors contributed equally to this work.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Only synthetic data were created.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Huang, K.; Wang, Z.A.; Song, C.; Xu, K.; Li, H.; Wang, Z.; Guo, Q.; Song, Z.; Liu, Z.B.; Zheng, D.; et al. Quantum generative adversarial networks with multiple superconducting qubits. *npj Quantum Inf.* **2021**, *7*, 165. [CrossRef]
2. Dallaire-Demers, P.L.; Killoran, N. Quantum generative adversarial networks. *Phys. Rev. A* **2018**, *98*, 012324. [CrossRef]
3. Chakrabarti, S.; Yiming, H.; Li, T.; Feizi, S.; Wu, X. Quantum Wasserstein generative adversarial networks. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019.
4. Kiani, B.T.; De Palma, G.; Marvian, M.; Liu, Z.W.; Lloyd, S. Learning quantum data with the quantum earth mover's distance. *Quantum Sci. Technol.* **2022**, *7*, 045002. [CrossRef]
5. Herr, D.; Obert, B.; Rosenkranz, M. Anomaly detection with variational quantum generative adversarial networks. *Quantum Sci. Technol.* **2021**, *6*, 045004. [CrossRef]
6. Situ, H.; He, Z.; Wang, Y.; Li, L.; Zheng, S. Quantum generative adversarial network for generating discrete distribution. *Inf. Sci.* **2020**, *538*, 193–208. [CrossRef]
7. Kalfon, B.; Cherkaoui, S.; Laprade, J.F.; Ahmad, O.; Wang, S. Successive Data Injection in Conditional Quantum GAN Applied to Time Series Anomaly Detection. *arXiv* **2023**, arXiv:2310.05307.
8. Ostaszewski, M.; Miszczak, J.; Banchi, L.; Sadowski, P. Approximation of quantum control correction scheme using deep neural networks. *Quantum Inf. Process.* **2019**, *18*, 126. [CrossRef]
9. Montanaro, A. Quantum algorithms: An overview. *npj Quantum Inf.* **2016**, *2*, 15023. [CrossRef]
10. LaPierre, R.; LaPierre, R. Shor algorithm. In *Introduction to Quantum Computing*; Springer: Cham, Switzerland, 2021; pp. 177–192.
11. Bhatia, V.; Ramkumar, K. An efficient quantum computing technique for cracking RSA using Shor's algorithm. In Proceedings of the 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 30–31 October 2020; pp. 89–94.
12. Roffe, J. Quantum error correction: An introductory guide. *Contemp. Phys.* **2019**, *60*, 226–245. [CrossRef]
13. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems 27 (NIPS 2014), Montreal, QC, Canada, 8–13 December 2014.
14. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 214–223.
15. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. Improved Training of Wasserstein GANs. *arXiv* **2017**, arXiv:1704.00028.
16. Ngo, T.A.; Nguyen, T.; Thang, T.C. A Survey of Recent Advances in Quantum Generative Adversarial Networks. *Electronics* **2023**, *12*, 856. [CrossRef]
17. Li, T.; Zhang, S.; Xia, J. Quantum generative adversarial network: A survey. *Comput. Mater. Contin.* **2020**, *64*, 401–438. [CrossRef]
18. Tsang, S.L.; West, M.T.; Erfani, S.M.; Usman, M. Hybrid Quantum–Classical Generative Adversarial Network for High-Resolution Image Generation. *IEEE Trans. Quantum Eng.* **2023**, *4*, 1–19. [CrossRef]
19. Goerg, G.M. The Lambert Way to Gaussianize heavy tailed data with the inverse of Tukey's h as a special case. *arXiv* **2012**, arXiv:1010.2265.
20. Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv* **2015**, arXiv:1505.00853.
21. Labiad, B.; Berrado, A.; Benabbou, L. Predicting extreme events in the stock market using generative adversarial networks. *Int. J. Adv. Intell. Inform.* **2023**, *9*, 218–230. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.