Physica A 654 (2024) 130140

Contents lists available at ScienceDirect

Physica A: Statistical Mechanics and its Applications

journal homepage: www.elsevier.com/locate/physa

Can Bitcoin trigger speculative pressures on the US Dollar? A novel ARIMA-EGARCH-Wavelet Neural Networks

David Alaminos [a,*], M. Belén Salas-Compás [b], Manuel Á. Fernández-Gámez [b]

[a] Department of Business, University of Barcelona, Av. Diagonal 690, Barcelona 08034, Spain
[b] Department of Finance and Accounting, University of Málaga, Campus de El Ejido 6, Málaga 29071, Spain

ARTICLE INFO

Keywords:
Cryptocurrency markets
Bitcoin
Volatility
ARIMA-EGARCH
Speculative attack
Short squeeze
Neural Networks
Wavelet analysis

ABSTRACT

In recent years, Bitcoin has garnered attention as a digital currency, prompting increasing debate regarding its effects on traditional financial markets, particularly the US dollar. This study investigates the relationship between Bitcoin and the US dollar, especially in the contexts of speculative attacks, where investors attempt to devalue a currency, and short squeezes, where rapid price rises force short sellers to quickly buy back assets to avoid further losses. The study employs a novel hybrid model combining an autoregressive moving average, Generalized Autoregressive Conditional Heteroskedasticity, and Wavelet Neural Networks techniques with neural networks approaches. The results suggest that significant trading activity in Bitcoin/US dollar, particularly during speculative attacks and short squeezes, can substantially impact the US dollar/EUR market, increasing price volatility as traders adjust their strategies. These adjustments, along with risk management strategies, drive higher trading volumes and further volatility. Our findings demonstrate that our novel hybrid model combined with Quantum Recurrent Neural Networks provides the most accurate predictions, offering valuable insights to inform trading strategies in both Bitcoin/US dollar and US dollar/EUR markets. This study has important implications for policymakers and market participants, emphasising the need to understand the relationship between Bitcoin and the US dollar for financial stability and effective policy formulation. It also highlights the necessity of advanced modeling techniques to accurately predict cryptocurrency market behavior.

1. Introduction

Technological advances have changed the way economic entities interact and influence market dynamics, leading to increased popularity and acceptance of digital currencies among financial institutions. Cryptocurrencies, created using cryptographic techniques, have transformed finance by operating independently of central banks and utilising blockchain technology [53]. Bitcoin is the most notable cryptocurrency, with a significant market capitalisation and a strong technological base, and represents a transformative innovation both in banking systems and in other aspects of the economy [88]. Notably, Bitcoin has changed how transactions, investments, and wealth storage are managed, offering greater transparency, lower fees, and faster cross-border transfers. Understanding their long-term implications is crucial, as cryptocurrencies also facilitate effective fundraising for socially beneficial projects [92].

However, while cryptocurrencies provide advantages like decentralisation and greater financial inclusion, they also pose

* Corresponding author.
E-mail address: alaminos@ub.edu (D. Alaminos).

https://doi.org/10.1016/j.physa.2024.130140
Received 13 August 2024; Received in revised form 27 September 2024; Accepted 2 October 2024
Available online 11 October 2024

considerable challenges that demand proper regulatory oversight to maintain economic stability. It is essential to implement suitable regulations to prevent fraud and safeguard consumers [11]. The growing adoption of Bitcoin may increase financial volatility and weaken the effectiveness of monetary policies. Jagtiani et al. [57] warn that its speculative nature can lead to bubbles and systemic risks. The lack of clear regulation hampers innovation, making international collaboration essential to establish regulations that balance the benefits and risks associated with cryptocurrencies. The possibility of implementing a central bank digital currency (CBDC) is an important and relevant topic of debate among economic experts and monetary authorities around the world. These digital representations of conventional currencies issued by governments, which are formally controlled and supervised by central authorities, have the potential to fundamentally change the nature of money in the digital age [24,82].

As cryptocurrencies, particularly Bitcoin, become significant financial assets, they have the potential to disrupt traditional monetary systems. Huang et al. [56] note that arbitrage strategies in Bitcoin markets impact US dollar exchange rates and liquidity. Investors exploit price differences across platforms to profit, causing fluctuations in the dollar's value. These activities boost dollar market liquidity but add volatility to Bitcoin prices, highlighting the interconnectedness of both markets.

In recent years, Bitcoin has emerged as a popular digital currency and an alternative to traditional fiat currencies such as the US dollar [13]. As of September 2021, Bitcoin's market capitalization exceeded $900 billion, making it one of the most valuable assets in the world [35]. Bitcoin's decentralized nature and limited supply have attracted investors seeking an alternative store of value and a hedge against inflation [40]. The US dollar, on the other hand, remains the world's dominant reserve currency, with its value impacting the global economy and financial markets [84]. The US dollar's strength and stability have made it a preferred currency for international trade and investment [49].

Given the growing importance of Bitcoin and its relationship with the US dollar, it is crucial to study the co-movements between Bitcoin and US dollar markets. The co-movements refer to the extent to which the prices of Bitcoin and the US dollar move together over time. Understanding the co-movements can provide insights into the potential spillover effects of Bitcoin on the US dollar and vice versa. One potential spillover effect is the possibility of a speculative attack or short squeeze. A speculative attack occurs when investors sell a currency in large volumes, causing its value to fall rapidly [66]. A short squeeze happens when investors who have bet against a currency are forced to buy it back at a higher price, driving up its value [93]. These effects can be exacerbated by the co-movements between Bitcoin and the US dollar.

In addition to shedding light on the potential adversarial effects of Bitcoin on the US dollar, this paper also aims to contribute to the literature on the co-movements between Bitcoin and traditional fiat currencies. Previous studies have examined the co-movements between Bitcoin and the US dollar [10,94,103,13,25,40,73,81]. Cao and Ling [25] analyze the asymmetric dependence between Bitcoin, the US dollar, and gold. Dyhrberg [40] highlights the importance of Bitcoin, gold, and the US dollar in investment portfolios and risk management. Szetela et al. [94] explore the mutual influence between Bitcoin and various currencies, including the US dollar. Antoniadis et al. [10] focus on Bitcoin's impact on the USDX, particularly the unequal relationship with the US dollar. Mokni and Ajmi [81] examine extreme Granger causality between cryptocurrencies and the US dollar before and after the pandemic. Ahmadova, Guliyev, and Aliyev [2] identify a long-term inverse relationship between the US dollar index and Bitcoin, confirmed by Granger causality. Liu et al. [73], however, argue that traditional cointegration and causality tests inadequately capture the dynamic interactions between Bitcoin and global financial assets, suggesting that these relationships may change across different economic phases. They note a cointegration between negative effects on the US dollar and positive effects on Bitcoin but find no causal link between gold or the US dollar and Bitcoin. Future research should explore the dynamic interactions between Bitcoin and global financial assets over varying timeframes. Wu [103] summarise that Bitcoin demonstrates attributes such as a highly speculative financial instrument, limited hedging efficacy under normal market conditions, and lacks the haven status during extreme market scenarios. Results from univariate Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models and the Vector Autoregressive (VAR) model indicate that Bitcoin exhibits a pronounced one-way spillover effect with USD/EUR, with no reciprocal spillover effect observed. Impulse response analyses indicate that Bitcoin is more responsive to external market shocks compared to the impact of Bitcoin shocks on other markets. They advocate for utilising high-frequency data in future investigations to delve into the relationship between Bitcoin and other conventional assets.

In previous years, more researchers have started using Wavelet Neural Networks (WNN) to create hybrid models for predicting time series from high-frequency financial data. The goal is to improve prediction accuracy. Ortega and Khashanah [85] propose a WNN model for short-term forecasting of stock market returns. They use high-frequency financial data to build this model. The hybrid approach combines wavelets and neural networks to capture the non-linear and non-stationary nature of financial time series. Their results show reasonable accuracy for predictions over one-, three-, and five-step horizons. Lee, Koh, and Choe [70] explore ways to enhance portfolio optimization, asset allocation, and trading systems. They apply deep reinforcement learning to achieve this. Their input data comes from high-frequency time series of S&P500, DJI, and KOSPI indices. These data are first decomposed using wavelet transform (WT). The authors conclude that wavelet-transformed high-frequency data provide valuable information for learning investment strategies.

Besides, in recent years, other authors have highlighted the benefits of hybrid WNN models. Huang and Wang (2018) developed a new hybrid model by combining discrete wavelet transform with a stochastic recurrent wavelet neural network. Their goal was to improve the accuracy of energy price predictions. Vogl, Rötzel, and Homes [98] demonstrated these hybrid models have advantages over traditional models in predicting financial markets. They conclude that the model can be a useful alternative to the sigmoid activation function. Wang, Sarker, and Bouri [101] used the WNN method to explore Bitcoin's macroeconomic dynamics. They considered factors like money supply, CPI, and EPU in the US. Their findings show that WNN is effective at analyzing temporal scales in the cryptocurrency market. It reveals time series patterns over different time frames, providing robust results. The main strength of WNN is its ability to capture time, location, and frequency information simultaneously. WNN is seen as a powerful tool for extracting

details and addressing past challenges faced by Convolutional Neural Networks (CNN) [75]. Yang and Wang [105] emphasize WNN's distinct advantages. Its basic components and overall architecture help it avoid blind spots, unlike backpropagation neural networks. Additionally, WNN has better learning capabilities and higher accuracy. Overall, its simple structure enables faster convergence and improved accuracy in learning tasks. Given the relative novelty of the cryptocurrency market and the limited literature on the potential impact of Bitcoin on the US dollar, alongside Bitcoin's unique characteristics compared to conventional assets from an econometric standpoint, there exists a gap in understanding [3]. To cover this gap, our study aims to provide valuable findings into the dynamics of the Bitcoin/US dollar market by exploring whether Bitcoin can instigate a speculative attack or trigger a short squeeze on the US dollar, thereby contributing to the advancement of knowledge in this area. Additionally, owing to the variety of trades and time scales in the cryptocurrency market, as discussed in the preceding paragraph, authors demonstrate the robust results that the hybrid WNN method provides [98,101,105]. Therefore, we apply the extension of Autoregressive Integrated Moving Average- Generalized Autoregressive Conditional Heteroskedasticity -Wavelet Neural Networks (ARIMA-GARCH-WNN). This method presents an innovative and powerful approach to estimating speculative attacks and short squeezes in the Bitcoin/USD market. Given the evolving nature of these phenomena and the unique characteristics of the cryptocurrency market, there is a pressing need for advanced modeling techniques. Speculative attacks and short squeezes in the Bitcoin/USD market are influenced by a wide range of factors, including market sentiment, investor behavior, regulatory changes, and macroeconomic events, making their accurate prediction a challenging task.

There are three main contributions of our research. First, by integrating ARIMA-GARCH models with WNN, we can overcome the limitations of traditional time series models and capture the intricacies of the Bitcoin/USD market. ARIMA-GARCH models provide a solid foundation for modeling the time-dependent dynamics and volatility clustering in the market. The incorporation of WNN enhances the model's ability to capture non-linear relationships and complex interactions, as they excel at processing multi-resolution and time-frequency information. Second, this paper can inform policymakers and market participants about the potential risks associated with Bitcoin's growing influence on the US dollar. The volatility and unpredictability of Bitcoin may pose challenges to financial stability and the functioning of global financial markets [67]. As such, policymakers may need to consider the implications of Bitcoin's co-movements with the US dollar when formulating monetary and regulatory policies. Third, this paper aims to make a significant contribution to the ongoing discussion surrounding the relationship between Bitcoin and the US dollar, specifically focusing on speculative attacks and short squeezes. By investigating the dynamics between these two currencies, the study seeks to provide valuable insights into the potential spillover effects of Bitcoin on the broader financial system. An essential aspect of the research is examining whether Bitcoin can introduce a speculative attack or short squeeze into the US dollar, and vice versa. Understanding these scenarios is crucial for policymakers and market participants as it sheds light on the risks and opportunities associated with Bitcoin's growing influence on the financial markets. The findings of this study will not only enhance our understanding of the intricate interplay between Bitcoin and the US dollar but also serve as a valuable resource for policymakers in managing potential risks and harnessing the opportunities presented by this evolving digital asset.

This article is structured as follows: Section 2 provides a comprehensive overview of the speculative experiences between Bitcoin and the US Dollar, setting the context for the study and describing the econometric setup used to estimate the data for speculative attacks and short squeezes models. Section 3 shows the model and analysis tools. In Section 4, we summarize the development of various algorithms, including ARMA-GARCH-WNN. Section 5 presents the results of the study and provides a detailed discussion. Section 6 indicates the discussion results. Lastly, in Section 7, the conclusions derived from the study are displayed.

2. Speculative movements in Bitcoin-USD: focus on attacks and squeezes

2.1. Speculative context in Bitcoin-USD market

Speculative attacks and short squeezes have been observed in the past in the Bitcoin and US dollar markets, where the co-movements between the two assets created opportunities for traders to exploit market inefficiencies. These events highlight the potential for Bitcoin to introduce adversarial co-movements in the Bitcoin/US dollar market.

The collapse of Mt. Gox, a major Bitcoin exchange, in 2014 is also considered a speculative attack on Bitcoin. The exchange suffered from a large-scale hack, resulting in the theft of hundreds of thousands of Bitcoins. This event caused the price of Bitcoin to plummet, and many investors who had been holding Bitcoins on the exchange suffered significant losses. The collapse of Mt. Gox also created uncertainty and distrust in the Bitcoin market, leading to a prolonged bear market that lasted for several years (Ishikawa, 2017).

There have been instances in the past where Bitcoin's co-movements with the US dollar have resulted in speculative attacks and short squeezes. For example, in December 2017, the price of Bitcoin reached an all-time high of nearly $20,000, triggering a speculative frenzy among investors. At the same time, the US dollar was experiencing a period of weakness due to concerns over the Trump administration's economic policies (Liu, 2018). As a result, some investors saw Bitcoin as a safe-haven asset and a hedge against the dollar, further fueling the speculative demand for Bitcoin.

However, as the price of Bitcoin continued to soar, it created a short squeeze in the futures markets, where investors who had bet against Bitcoin were forced to buy back their positions to limit their losses (Cochrane, 2017). This short squeeze further intensified the demand for Bitcoin and led to a sharp increase in its price. The co-movements between Bitcoin and the US dollar during this period illustrate the potential for Bitcoin to introduce a speculative attack or short squeeze into the dollar.

Another example of Bitcoin's impact on the US dollar occurred in March 2020, during the COVID-19 pandemic. As fears over the pandemic's economic impact grew, investors rushed to safe-haven assets, including Bitcoin and the US dollar. However, as the Federal Reserve implemented monetary easing policies, including cutting interest rates to near zero and injecting trillions of dollars into the financial system, the US dollar weakened significantly. This weakness, in turn, contributed to a surge in the price of Bitcoin, which

reached a yearly high of nearly $65,000 in April 2021 (Bloomberg, 2021).

Conversely, a speculative attack on Bitcoin was observed in early 2021 when the price of Bitcoin surged to nearly $65,000. This surge was accompanied by a sharp appreciation of the US dollar, which had been strengthening due to the rollout of COVID-19 vaccines and expectations of a strong economic recovery (Bloomberg, 2021). Some investors saw the US dollar as a more stable asset compared to Bitcoin and shorted Bitcoin to bet against its price increase. This created a short squeeze where investors who had bet against Bitcoin were forced to buy back their positions to limit their losses, further intensifying the demand for Bitcoin and contributing to its price increase.

Another example of a speculative attack in the Bitcoin market was observed in 2013 when the price of Bitcoin surged from around $13 to over $1000 within a few months. This surge was accompanied by a sharp depreciation of the US dollar, which was experiencing a period of low interest rates and quantitative easing by the Federal Reserve (Gandal et al., 2018). Some investors saw Bitcoin as an alternative investment opportunity and a hedge against inflation, further fueling the speculative demand for Bitcoin. This created a feedback loop where the depreciation of the dollar fueled demand for Bitcoin, and the increase in the price of Bitcoin further weakened the dollar.

These past events illustrate the potential for Bitcoin to introduce speculative attacks and short squeezes into the US dollar and vice versa. By examining the co-movements between Bitcoin and the US dollar, this paper seeks to provide further insights into the potential risks and opportunities associated with Bitcoin's impact on the broader financial system. There is a need for further research into the co-movements between Bitcoin and traditional currencies such as the US dollar.

2.2. Interpretation of speculative attacks and short squeezes observations

In this section, we delve into the dynamics of speculative attacks and short squeezes, two significant phenomena that influence financial markets. Speculative attacks involve aggressive selling or shorting of assets to drive down their prices, often leading to market instability and investor losses. On the other hand, short squeezes occur when a sharp price surge forces short sellers to cover their positions, further accelerating the price increase.

2.2.1. Speculative attacks

A speculative attack occurs when speculators aggressively sell or short a specific asset to drive down its price. The primary outcome of a speculative attack is a decline in the targeted asset's price. This depreciation can have various consequences, including devaluing the asset and causing potential losses for investors. Additionally, a significant and prolonged decrease in price can undermine market confidence and intensify the downward pressure on the asset. The fundamental interpretation of a speculative attack revolves around intentional and aggressive actions taken to reduce the price of the targeted asset [65].

In our study, we employ the well-established framework proposed by Glick and Moreno (1999) to analyze and estimate the occurrences of speculative attacks on currencies. Glick and Moreno's model of currency crises has been widely acknowledged and extensively utilized in the field of international finance. By adopting this robust framework, we aim to gain insights into the dynamics and potential triggers of speculative attacks, which play a significant role in shaping the stability and volatility of global financial markets.

Glick and Moreno (1999) devised a method to determine exchange rate crises in East Asian and Latin American countries during the period from January 1972 to October 1997. They constructed an index by examining the percentage rate of change in real exchange rates. Their rationale was based on the argument that when the real exchange rate appreciates significantly, it elevates the probability of a currency crisis occurring. By focusing on the percentage rate of change in real exchange rates, Glick and Moreno aimed to capture the magnitude and direction of currency fluctuations over time. They believed that substantial increases in the real exchange rate appreciation could serve as a warning sign, indicating a higher likelihood of an imminent currency crisis.

$$Speculative\ Attacks = \frac{\Delta RER}{\Delta t} > k \tag{1}$$

$\frac{\Delta RER}{\Delta t}$ denotes the rate of change of the real exchange rate over time, k is a threshold value, and when the rate of change exceeds this threshold, it may be considered a speculative attack. So, for Bitcoin/USD with a standard deviation of 6.2 % in daily retorns (using historical data from CoinMarketCap: https://coinmarketcap.com), a threshold k of 9.3 % would be appropriate if we apply a multiplier of 1.5. This threshold implies that any daily change in the real exchange rate greater than 9.3 % could be considered significant enough to potentially indicate a speculative attack, according to the framework proposed by Glick and Moreno (1999).

To model speculative attacks as per Glick and Moreno (1999) using ARMA-EGARCH, the equations as follows:

Equation for ΔRER (Rate of Change in Real Exchange Rates):

$$\Delta RER_t = \mu + ARMA_{lags} + \varepsilon_t \tag{2}$$

Equation for Conditional Variance (σ_t^2) using EGARCH:

$$\sigma_t^2 = \omega + \alpha(\varepsilon_{t-1})^2 + \beta\sigma_{t-1}^2 \tag{3}$$

where ΔRER_t represents the rate of change in real exchange rates at time t, $ARMA_{lags}$ represents the AutoRegressive and Moving Average terms in the ARMA process for ΔRER, ε_t represents the error term at time t, σ_t^2 represents the conditional variance at time t, and ω, α, and β are parameters to be calculated.

Calculating cryptocurrency crises solely based on the real exchange rate can be challenging due to the unique characteristics and complexities of the cryptocurrency market. However, there have been instances where the real exchange rate has provided insights into potential crises or market disruptions.

Bitcoin and the Mt. Gox incident revealed the vulnerability of cryptocurrencies to security breaches, as the loss of a substantial number of Bitcoins caused a significant depreciation in the real exchange rate of Bitcoin. This depreciation reflected a crisis of confidence in Bitcoin and raised concerns about the overall security and stability of the cryptocurrency market. China's regulatory crackdown on cryptocurrencies, specifically the ban on ICOs and trading activities, had a profound impact on the real exchange rate of cryptocurrencies like Bitcoin and Ethereum. The resulting depreciation indicated a crisis driven by regulatory actions, reflecting market uncertainty and apprehension regarding the future of cryptocurrencies in China. The real exchange rate has played a crucial role in identifying periods of market corrections and speculative bubbles in the cryptocurrency realm. During speculative bubbles, the real exchange rate of Bitcoin, for example, surged to record highs before experiencing a sharp decline. This depreciation signaled the bursting of the bubble and marked the onset of a crisis period for Bitcoin and the wider cryptocurrency market [102].

2.2.2. Short squeezes

A short squeeze is commonly observed in the stock market when a stock's price undergoes a rapid surge. This situation arises when there is significant short-selling activity for a particular stock, and subsequently, an abrupt surge in demand or positive news triggers a sharp rise in the stock price. Consequently, individuals holding short positions (anticipating a decline in stock price) may be compelled to buy the stock to close their positions, resulting in further upward pressure on the price. Consequently, a short squeeze frequently results in a swift escalation of stock prices [8].

Initially, we employ standard tests to evaluate whether the returns exhibit a random distribution and adhere to the random walk hypothesis, as proposed by Fama (1965) and Malkiel (1973). Furthermore, we propose that if the leverage effect is considered a typical characteristic of financial time series, the emergence of the opposite phenomenon during a short-squeeze period provides quantitative evidence of abnormality.

Modeling a short squeeze between Bitcoin and the US dollar involves analyzing various factors and market dynamics. While there isn't a single equation to capture all aspects, we can provide a simplified equation that takes into account some key components [54]:

$$Short\ Interest\ Ratio(SIR) = \frac{Total\ Short\ Interest(TSI)}{Average\ Daily\ Trading\ Volume(ADTV)} \tag{4}$$

where:

- **Short Interest Ratio (SIR)**: This ratio indicates the level of short interest relative to the average daily trading volume. A high SIR suggests a higher potential for a short squeeze.
- **Total Short Interest (TSI)**: The total number of Bitcoin positions held in a short position. This can be calculated based on exchange data.
- **Average Daily Trading Volume (ADTV)**: The average volume of Bitcoin traded daily. This provides context for how quickly short positions can be covered.

To model short squeezes involving Bitcoin and US dollar using ARMA-EGARCH, you can adapt the equations as follows:
Equation for SIR (Short Interest Ratio):

$$SIR_t = \mu + ARMA_{lags} + \varepsilon_t \tag{5}$$

Equation for Conditional Variance (σ_t^2) using EGARCH:

$$\sigma_t^2 = \omega + \alpha(\varepsilon_{t-1})^2 + \beta\sigma_{t-1}^2 \tag{6}$$

Various tests can be applied to determine whether the returns are randomly distributed and consistent with the Efficient Market Hypothesis (EMH). These include Serial Correlation tests, Unit Root tests, Multiple Variance Ratio analysis, the Adaptive Market Hypothesis, and Runs-tests (e.g., Borges (2010), Lo (2004), Worthington and Higgs (2004)). The selection of tests is guided by the characteristics of the time series under investigation, such as whether they are parametric or non-parametric. In this study, we employ the Runs-test to assess if the returns during the short squeeze period exhibit a random distribution.

A 'run' is defined as a sequence of consecutive positive or negative returns, and the expected number of runs (μ_R) can be calculated using the formula: $\mu_R = 2N_P N_n/N+1+1/N$, where N represents the total observations, and N_P and N_n are the numbers of positive and negative observations, accordingly. Deviations from this expected number indicate potential patterns in price behavior, such as reversals or sustained growth, and allow us to assess the randomness of returns using the Z-statistic [97].

According to the Efficient Market Hypothesis (EMH), which posits that asset prices reflect all available information, short-term returns should exhibit a random distribution between positive and negative values on an hour-to-hour basis. A "run" is documented when consecutive positive or negative returns occur. The expected number of runs can be calculated using a specific formula [97].

$$Z = \frac{R - \mu_R}{\sigma_R} \tag{7}$$

An unusually high number of runs indicates a pattern where prices initially increase (or decrease) but then reverse direction,

showing a decline (or rise). On the other hand, a lower-than-expected number of runs suggests a phase of growth or decline in which stock market prices do not immediately reflect all available information. Moreover, if the returns from the previous hour are positive (or negative), there is a likelihood that the current hour's returns will follow the same trend. To quantitatively evaluate how closely the number of runs aligns with expectations and to determine if the yields are drawn in a random distribution, we employ the Z-statistic [97].

3. Modelling and analysis tools

In recent years, advanced computational techniques have emerged as promising tools for analysing contagion dynamics. We propose a novel methodology that combines the strengths of ARIMA, EGARCH, and wavelet neural networks to study contagion measures comprehensively. This work employs the model introduced by Diebold and Yilmaz [38] to assess the contagion effect. Utilizing this approach, we analyse the dynamic and conditional spillover to track the progression of Bitcoin's contagion impact. Additionally, we employ the two-period unconditional spillover index to contrast the contagion levels between Bitcoin and alternative financial markets. By adopting this methodology, we derive directional outcomes within a generalized VAR (Vector Autoregression) approach. This framework effectively resolves the issue of order dependency encountered in variance decompositions, as highlighted by Diebold and Yilmaz [38,37].

The moving average procedure for VAR(p) is shown by

$$y_t = \sum_{i=0}^{\infty} A_i \varepsilon_{t-i} \tag{8}$$

where A_i is N × N coefficient matrix in recursive pattern.

The model computes the variance decompositions for H-step-ahead forecast errors denoted by $\theta_{ij}^{g}(H)$, where H=1,2,3,…

$$\theta_{i\leftarrow j}^{g}(H) = \frac{\sigma_{jj}^{-1} \sum_{h=0}^{H-1} \left(e_i A_h \sum e_j\right)^2}{\sum_{h=0}^{H-1} \left(e_i A_h \sum A_h^{`} e_i\right)} \tag{9}$$

In this context, $\theta_{i\leftarrow j}^{g}(H)$ denotes the transmission of risk transmission from asset j to asset i. The matrix Σ represents the variance of the error vector ε, where σ_{jj} denotes the standard deviation of the error term for asset j. e_i and e_j are selection vectors of size N × 1, where only the i-th element is set to one, and the rest are zeros. Notably, the shocks to each variable are not orthogonalized, meaning the sum of contributions in each row may not necessarily total one. For the sake of comparison, a normalized variance decomposition matrix is provided [90]:

$$\widetilde{\theta}_{i\leftarrow j}^{g}(H) = \frac{\theta_{ij}^{g}(H)}{\sum_{j=1}^{H} \theta_{ij}^{g}(H)} \tag{10}$$

where, $\sum_{i,j=1}^{N} \widetilde{\theta}_{i,j}^{g}(H) = N$ and $\sum_{j=1}^{N} \widetilde{\theta}_{i,j}^{g}(H) = 1$

The market i's directional spill-over to market j is expressed by

$$S_{\bullet i}^{g}(H) = \frac{\sum_{j=1, i\neq j}^{N} \widetilde{\theta}_{ji}^{g}(H)}{\sum_{i,j=1}^{N} \widetilde{\theta}_{ji}^{g}(H)} X100 = \frac{\sum_{j=1, i\neq j}^{N} \widetilde{\theta}_{ji}^{g}(H)}{N} X100 \tag{11}$$

The indirect effect on market i from market j is given by

$$S_{i\bullet}^{g}(H) = \frac{\sum_{j=1, i\neq j}^{N} \widetilde{\theta}_{ij}^{g}(H)}{\sum_{i,j=1}^{N} \widetilde{\theta}_{ij}^{g}(H)} X100 = \frac{\sum_{j=1, i\neq j}^{N} \widetilde{\theta}_{ij}^{g}(H)}{N} X100 \tag{12}$$

And the net spillover equals to

$$S_{i}^{g}(H) = S_{\bullet i}^{g}(H) - S_{i\bullet}^{g}(H) \tag{13}$$

This study employs net pairwise spillover to illustrate the contagion pathway. Net pairwise spillover characterizes the overall contagion impact between two markets or assets, aiding in the identification of Bitcoin's role to other financial markets and assets in a directional manner. The net pairwise spillover is calculated as follows [90]:

$$S_{ij}^{g}(H) = \frac{\widetilde{\theta}_{ji}^{g}(H)}{\sum_{i,k=1}^{H} \widetilde{\theta}_{ik}^{g}(H)} - \frac{\widetilde{\theta}_{ij}^{g}(H)}{\sum_{j,k=1}^{H} \widetilde{\theta}_{jk}^{g}(H)} X100 = \frac{\widetilde{\theta}_{ji}^{g}(H) - \widetilde{\theta}_{ij}^{g}(H)}{N} X100 \tag{14}$$

On another note, network theory has been extensively employed in studies concerning financial contagion and market interconnectedness [39,45,46,77]. Building on the contagion measures introduced by Diebold and Yilmaz [38], we integrate the network structure topology to conceptualise and elucidate the Bitcoin and other financial market contagion networks. In the analysis of financial contagion networks, each financial institution is represented as a node, with links denoting the connections among various entities. These links are addressed and weighted, capturing the display of individual organisations [45].

To illustrate the contagion and connections between Bitcoin and other financial markets, four common measurements are employed. Firstly, the degree of a node, which signifies the number of connections to additional nodes, is calculated as follows: $d_i = \sum_{j=1}^{n} a_{ij} = \sum_{j=1}^{n} a_{ji}$. The in-degree measures the number of incoming connections to the node $d_{in}(i) = \sum_{j=1}^{n} a_{ij}$, while the out-degree represents the number of outgoing connections from the node $d_{out}(i) = \sum_{j=1}^{n} a_{ij}$. Secondly, betweenness centrality assesses the control a node exerts over the dissemination of risk and information throughout the network [43,44]. It is computed as the fraction of shortest paths between pairs of nodes that traverse the node of interest: $BC_i = \sum_{s \neq i \neq j} \frac{n_{st}^{i}}{g_{st}}$. Finally, the average path length and average diameter are used to depict the overall changes in the contagion network both before and during the COVID-19 time. The average path length refers to the mean shortest distance between all pairs of nodes iii and jjj in the network, while the average diameter represents the typical longest distance between any two nodes. The diameter is computed as $s_{\max} = \max_{i,j} s_{i,j}$.

3.1. ARMA-EGARCH model equations

The ARMA-EGARCH model captures the autoregressive and moving average behavior of the time series along with the conditional volatility dynamics. The equations for the ARMA-EGARCH model are as follows [113]:

ARMA Equations:

$$Y_t = c + \sum_{i=1}^{p} \varnothing_i \bullet Y_{t-i} + \sum_{i=1}^{q} \theta_j \bullet \varepsilon_{t-j} + \varepsilon_t \tag{15}$$

EGARCH Equations:

$$\log(\sigma_t^2) = \omega + \alpha \bullet |\varepsilon_{t-1}| + \sum_{i=1}^{q} \beta_i \bullet \log(\sigma_{t-i}^2) + \gamma \bullet \varepsilon_{t-1} \tag{16}$$

being Y_t is the observed value at time t, c is a constant term, $\varnothing_i$ represents the autoregressive coefficients for lags 1 to p, θ_j represents the moving average coefficients for lags 1 to q, ε_t is the error term at time t, $\log(\sigma_{t-i}^2)$ is the logarithm of the conditional variance at time t, ω is the constant term. In the same way, α represents the parameter controlling the asymmetric effects of positive and negative shocks, β_i represents the parameters controlling the persistence of the conditional variance, and γ represents the parameter capturing the leverage effect.

3.2. Wavelet Neural Networks framework

Wavelet networks represent a novel class of networks that integrate traditional sigmoid neural networks (NNs) with wavelet analysis (WA). WNs have been highly successful across a broad variety of activities. WA has proven to be an effective tool for analysing various time series and has been successfully applied in fields such as “image processing”, “signal de-noising”, “density estimation”, “signal and image compression”, and “time-scale decomposition”. WA is often considered a "microscope" in mathematics [23] and is a powerful method for representing nonlinearities (Fang & Chow, 2006). However, a significant limitation of WA is its restriction to applications with small input dimensions, as constructing a wavelet basis becomes computationally demanding when the input vector’s dimensionality is relatively high [109].

Conversely, NNs can approximate any deterministic nonlinear process with minimal prior knowledge and without making assumptions about the process. However, classical sigmoid NNs come with certain limitations. Typically, the initial weights in an NN are assigned randomly, leading to longer training times. Additionally, when the transfer function is sigmoidal, there is a considerable risk that the training algorithm will get stuck in local minima. At last, there is no theoretical connection between the specific parameterization of a sigmoidal activation function and the optimal network framework, which is not the case with wavelet networks (WNs) [91].

In their 1993 study, Pati and Krishnaprasad demonstrated that feedforward NNs could be theoretically formulated using wavelet decompositions. Zhang and Benveniste [110] introduced WNs as an alternative to feedforward NNs to overcome the limitations of traditional methods. WNs generalise radial basis function networks (RBFNs) and consist of a single hidden layer that uses wavelets as activation functions instead of the conventional sigmoidal functions. Notably, WNs retain the "universal approximation" property of NNs. The nodes, or "wavelons," represent significant wavelet coefficients in the function’s expansion. Bernard, Mallat, and Slotine [19] highlighted the advantages of using wavelets over other transfer functions, citing their high compression capabilities and the efficiency

of updating function estimates with only a small subset of coefficients.

This section outlines the structure of a WN, which typically consists of three layers: an input layer, a hidden layer, and an output layer. The input layer introduces the explanatory variables to the WN. The hidden layer, composed of hidden units (HUs) known as wavelons, functions similarly to neurons in traditional sigmoid neural networks. In this layer, the input variables are transformed into dilated and translated versions of the mother wavelet. Finally, the output layer estimates the approximation of the target values.

The concept behind a WN is to adapt the wavelet basis to the training data, making it potentially more efficient than a sigmoid NN [107]. Various adaptations of WNs have been proposed. Billings and Wei [20], Kadambe and Srinivasan [59], Mellit et al. [79], and Xu and Ho [104] used adaptive WNs. Chen et al. [30] introduced a local linear WN, where the connection weights between the hidden and output layers are replaced by a local linear model. Fang and Chow (2006) and Jiao et al. [58] proposed a multiwavelet NN, which uses a linear combination of wavelet bases as the activation function, with all wavelet weights updated during training. This structure, enhanced by the discrete wavelet transform (DWT), improves the network's approximation capabilities. Khayamian et al. [61] introduced a principal component-wavelet NN, combining principal component analysis (PCA) for dimensionality reduction with a WN for function approximation. Zhao et al. [111] employed a multidimensional wavelet-basis function NN, using a wavelet function in the hidden layer and a sigmoid function in the output layer. Becerikli [15] proposed a dynamic WN with unconstrained connectivity and dynamic elements, incorporating lag dynamics in its wavelet processing units.

We utilise a multidimensional WN featuring a linear link between the wavelons and the output. To ensure the model performs effectively when linearity is present, we include direct links from the input layer to the output layer. Consequently, if the network has no HUs it simplifies to a linear model.

The next equation represents the network output:

$$g_{\lambda}(x;w) = \widehat{y}(x) = \omega_{\lambda+1}^{[2]} + \sum_{j=1}^{\lambda} \omega_j^{[2]}.\psi_j(x) + \sum_{i=1}^{m} \omega_i^{[0]}.x_i \tag{17}$$

where $\psi_j(x)$ denotes a multidimensional wavelet built by the product of m scalar wavelets, x represents the input vector, m shows the number of network inputs, λ indicates the number of HUs and w is a network weight [108]. The multidimensional wavelets are calculated according to the following equation, being ψ the mother wavelet:

$$\psi_j(x) = \prod_{i=1}^{m} \psi z_{ij} \tag{18}$$

$$z_{ij} = \frac{x_i - \omega_{(\xi)ij}^{[1]}}{\omega_{(\zeta)ij}^{[1]}} \tag{19}$$

being i = 1, …, m, j = 1, …, λ + 1 and the weights ω denotes the translation ($\omega_{(\xi)ij}^{[1]}$) and the dilation ($\omega_{(\zeta)ij}^{[1]}$) factors. The full network parameters vector are: $\omega = \left(\omega_i^{[0]}, \omega_j^{[2]}, \omega_{\lambda+1}^{[2]}, \omega_{(\xi)ij}^{[1]}, \omega_{(\zeta)ij}^{[1]}\right)$.

The choice of the mother wavelet varies depending on the specific application and is not restricted to the options mentioned. The activation function can be a wavenet (using orthogonal wavelets) or a wave frame (using continuous wavelets). In line with the work of Becerikli, Oysal, and Konar [16], Billings and Wei [20], and Zhang [108], we use the Mexican Hat function as the mother wavelet, as it has proven effective and reliable in various applications. This function is defined by:

$$\psi(z_{ij}) = (1 - z_{ij}^2)e^{-1/2 z_{ij}^2} \tag{20}$$

3.3. Network parameter initialisation

In WNs, unlike in NNs that employ sigmoid functions, randomly selecting starting values for dilation and translation parameters may be ineffective [86]. Because a wavelet is a waveform of limited duration with an medium value of zero and localized properties, random initialization can result in wavelons with zero values. Training algorithms like gradient descent with random initialization are not efficient [107] since random initialization can slow down training and lead to a local minimum in the loss function [87]. Additionally, in sigmoid NNs, while minimizing the loss function can be achieved with random initialization, the resulting weights can change from time to time [9].

By leveraging information from the wavelet analysis (WA) of the input dataset, the initial values of the network parameters can be selected more efficiently. Efficient initialization reduces the number of training iterations and helps training algorithms avoid local minima in the loss function. Moreover, it ensures that the weights minimizing the loss function are consistently approximated. Some approacges have been suggested to optimize the initialization of wavelet parameters. For example, Zhang and Benveniste [110] introduced a specific initialization approach for the translation and dilation parametres as follows.

$$\omega_{(\xi)ij}^{[1]} = 0.5 \quad (N_i + M_i) \tag{21}$$

$$\omega_{(\zeta)ij}^{[1]} = 0.2 \quad (M_i - N_i) \tag{22}$$

where $M_i$and $N_i$are defined as the maximum and minimum of input x_i.

$$M_i = \max_{p=1,\ldots,n} \quad (X_{ip}) \tag{23}$$

$$N_i = \min_{p=1,\ldots,n} \quad (X_{ip}) \tag{24}$$

The direct connections $\omega_i^{|0|}$ and weights $\omega_j^{|2|}$ are initialized with small random values between 0 and 1, as their precise initialization is less critical. While this heuristic approach is straightforward and has minimal computational cost, it is not particularly efficient. It does not guarantee finding the global minimum during training and fails to leverage the valuable information provided by wavelet decomposition [86].

4. Methodology

4.1. ARMA-EGARCH-Wavelet Neural Networks-Recurrent Neural Networks (ARMA-EGARCH-WNN-RNN)

RNNs have demonstrated success in various domains for time-series prediction, largely owing to their significant predictive capabilities. The typical structure of an RNN relies on its output, which is intricately linked to its prior computations [100]. When presented with an input sequence vector, the hidden states of a recurrent layer ss and the output of a single hidden layer y can be computed as shown in Eqs. (25) and (26).

$$s_t = \sigma(W_{xs}x_t + W_{ss}s_{t-1} + b_s) \tag{25}$$

$$y_t = o(W_{so}s_t + b_y) \tag{26}$$

where W_{xs}, W_{ss}, and W_{so} represent the connections from the input layer x to the hidden layer s, from the hidden layer to itself, and from the hidden layer to its output layer, respectively. Additionally, b_y denotes the biases asspciated with both the hidden layer and the output layer. σ and o are the activation functions highlighted in the equation.

$$STFT\{z(t)\}(\tau,\omega) = \int_{-\infty}^{+\infty} z(t)\omega(t-\tau)e^{-j\omega t}dt) \tag{27}$$

here $z(t)$ represents the vibration signals, $\omega(t)$ denotes the Gaussian window function centred around 0. Furthermore, $T(\tau, \omega)$ stands as a complex function describing the vibration signals across both time and frequency domains. To calculate the hidden layers with the convolutional operation, we apply Eqs. (28) and (29).

$$S_t = \sigma(W_{TS} * \quad T_t + W_{SS} * \quad S_{t-1} + B_s) \tag{28}$$

$$Y_t = o(W_{YS} * \quad S_t + B_y) \tag{29}$$

In the transition from inputs to hidden layers, convolution is executed between the weights and inputs, denoted by the term W, which represents the convolution kernels.

$$S_t = \sigma(W_{xs}\times x_t + W_{ss} \times s_{t-1} + W_{ar}\times r_{t-1} + W_{\omega s}\times \omega_{t-1} + b_s) \tag{30}$$

$$Y_t = o(W_{so} \times \quad S_t + B_y) \tag{31}$$

In the integrated model, the hidden state at time step s_t is computed using the activation function σ applied to the weighted sum of various inputs and biases. These inputs include the current input vector x_t, weighted by W_{xs}, which captures the influence of the current input; the previous hidden state s_{t-1}, weighted by W_{ss}, which integrates information from the previous time step and maintains temporal dependencies; the output of the ARMA-EGARCH model r_{t-1}, weighted by W_{ar}, which incorporates conditional mean and volatility information; and the output of the Wavelet Neural Network (WNN) w_{t-1}, weighted by W_{ws}, which includes features extracted through wavelet analysis. Additionally, a bias term b_s is included to adjust the weighted sum before applying the activation function. The final output y_t of the Recurrent Neural Network (RNN) is obtained by applying an activation function ω to the weighted sum of the hidden state s_t, weighted by W_{so}, and a bias term B_y [6].

4.2. ARMA-EGARCH-Wavelet Neural Networks-Convolutional Neural Network (ARMA-EGARCH-WNN-CNN)

In 1998, Lecun et al. introduced CNN as a network model. It is a type of feed-forward neural network renowned for its effectiveness in both image and natural language processing [62]. CNN has shown promise in time-series forecasting as well. One of its key advantages lies in its ability to locally sense data and distribute weights, resulting in a significant reduction in parameter numbers and an increase in learning efficiency [89]. Comprising primarily of two components, CNN includes the convolution layer and the clustering layer. The convolution layer uses multiple convolution kernels, as described in Eq. (32). While it efficiently extracts data features, it often results in high-dimensional feature maps. To reduce dimensionality and lower training costs, a clustering layer is added directly after the convolution layer.

$$l_t = ReLU(x_t \times k_t + b_t) \tag{32}$$

In the context of convolution, l_t denotes the output value, where t_{nh} represents the activation function. The input vector is denoted by x_t, while k_t signifies the weight of the convolution kernel, and b_t represents the bias associated with the convolution kernel. The combination of ARMA-EGARCH-WNN-CNN methodologies represents a novelty in time series forecasting, where intricate dependencies and patterns within data streams are meticulously captured and leveraged for predictive analytics. However, its utility in time-series forecasting is equally promising, with its innate ability to discern local data patterns and distribute weights efficiently [62]. As a novel fusion, the ARMA-EGARCH-WNN-CNN model not only inherits the robust forecasting capabilities of ARMA and EGARCH but also integrates the feature extraction prowess of WNN and the spatial awareness of CNN. This new development is defined by the next equation.

$$l_t = ReLU(x_t \times k_t + b_t + W_{ar} \times r_{t-1} + W_{\omega s} \times \omega_{t-1}) \tag{33}$$

In the ARMA-EGARCH-WNN-CNN model, l_t represents the output value at time step t after applying the convolution and activation. The activation function used is ReLU. The input vector at time step t is denoted by x_t. The weight of the convolution kernel is represented by k_t, and the bias associated with the convolution kernel is b_t. The weight matrix W_{ar} connects the ARMA-EGARCH model output r_{t-1} to the convolutional output, where r_{t-1} is the output of the ARMA-EGARCH model at the previous time step, representing conditional mean and volatility. Additionally, the weight matrix $W_{\omega s}$ connects the WNN output ω_{t-1} to the convolutional output, where ω_{t-1} is the output of the WNN at the previous time step, representing features extracted through wavelet analysis.

4.3. ARMA-EGARCH-Wavelet Neural Networks-Long Short-Term Memory (ARMA-EGARCH-WNN-LSTM)

LSTM, an advanced iteration of recurrent neural network (RNN), is tailored to handle time series data with time-dependent variables [55]. This architecture offers a significant advantage by assimilating historical information, enabling more accurate predictions of future states, particularly when the input data exhibits dependencies.

In RNN, the consideration of memory effect involves the utilization of a loop cell, facilitating the incorporation of past information to predict subsequent steps. However, the conventional structure of RNN lacks efficacy in processing long-term dependencies due to the occurrence of vanishing gradients during backpropagation. To surmount this challenge, LSTM networks have been devised, featuring a more efficient architecture comprising three distinct gates: "input gates, output gates, and forget gates". These gates ensure the retention of previous information while ensuring stable gradient computation. Within the LSTM cell state, information undergoes sequential processing as per the equations outlined in (41) [42,55].

$$\begin{aligned}
\boldsymbol{i}_t &= \boldsymbol{\sigma}(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \\
\boldsymbol{f}_t &= \boldsymbol{\sigma}(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{U}_f \boldsymbol{h}_{t-1} \quad + \quad \boldsymbol{b}_f) \\
\boldsymbol{o}_t &= \boldsymbol{\sigma}(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{U}_o \boldsymbol{h}_{t-1} \quad + \boldsymbol{b}_o) \\
\boldsymbol{g}_t &= \boldsymbol{tanh}(\boldsymbol{W}_c \boldsymbol{x}_t + \boldsymbol{U}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c) \\
\boldsymbol{C}_t &= \boldsymbol{f}_t \quad \otimes \quad \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \quad \otimes \quad \boldsymbol{C}_t \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \quad \otimes \boldsymbol{tanh}(\boldsymbol{C}_t)
\end{aligned} \tag{34}$$

In this system of equations, we develop into the dynamics of the ARMA-EGARCH-WNN-LSTM model, an integrated framework designed to address the complexities of time series analysis and forecasting. The equations encapsulate the intricate interplay between various components, including the Autoregressive Moving Average (ARMA) terms capturing historical dependencies, the Exponential Generalized Autoregressive Conditional Heteroskedasticity (EGARCH) terms modeling volatility clustering and asymmetry, the Wavelet Neural Network (WNN) terms uncovering additional patterns, and the Long Short-Term Memory (LSTM) network enabling the retention of long-term dependencies, expressed in the next equations:

$$\begin{aligned}
\boldsymbol{i}_t &= \boldsymbol{\sigma}(\boldsymbol{W}_{ix} \times \boldsymbol{x}_t + \boldsymbol{U}_{ih} \times \boldsymbol{h}_{t-1} + \boldsymbol{W}_{ir} \times \boldsymbol{r}_{t-1} + \boldsymbol{W}_{i\omega} \times \boldsymbol{\omega}_{t-1} + \boldsymbol{b}_i) \\
\boldsymbol{f}_t &= \boldsymbol{\sigma}(\boldsymbol{W}_{fx} \times \boldsymbol{x}_t + \boldsymbol{U}_{fh} \times \boldsymbol{h}_{t-1} + \boldsymbol{W}_{fr} \times \boldsymbol{r}_{t-1} + \boldsymbol{W}_{f\omega} \times \boldsymbol{\omega}_{t-1} + \quad \boldsymbol{b}_f) \\
\boldsymbol{o}_t &= \boldsymbol{\sigma}(\boldsymbol{W}_{ox} \times \boldsymbol{x}_t + \boldsymbol{U}_{oh} \times \boldsymbol{h}_{t-1} \quad + \boldsymbol{W}_{o\omega} \times \boldsymbol{\omega}_{t-1} + \boldsymbol{b}_o) \\
\boldsymbol{g}_t &= \boldsymbol{tanh}(\boldsymbol{W}_{gx} \times \boldsymbol{x}_t + \boldsymbol{U}_{gh} \boldsymbol{h}_{t-1} + \boldsymbol{W}_{g\omega} \boldsymbol{\omega}_{t-1} + \boldsymbol{b}_g) \\
\boldsymbol{C}_t &= \boldsymbol{f}_t \quad \otimes \quad \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \quad \otimes \quad \boldsymbol{g}_t \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \quad \otimes \boldsymbol{tanh}(\boldsymbol{C}_t)
\end{aligned} \tag{35}$$

In the integration of the ARMA-EGARCH model and Wavelet Neural Networks (WNN) with Long Short-Term Memory (LSTM) networks for time series forecasting, various components are defined as follows: i_t, f_t, o_t, and g_t represent the input, forget, output, and

cell gate activations, respectively. x_t represents the input at time step t, h_{t-1} represents the hidden state at the previous time step, r_{t-1} represents the output of the ARMA-EGARCH model at the previous time step, and ω_{t-1} represents the output of the WNN at the previous time step. The sigmoid activation function σσ and hyperbolic tangent activation function *tanh* are utilized, while $W_{\cdot x}$, $U_{\cdot h}$, $W_{\cdot r}$, and $W_{\cdot \omega}$ denote weight matrices connecting the input, hidden state, ARMA-EGARCH output, and WNN output, respectively, to the gates. Additionally, b. are bias vectors associated with the gates, and ⊙ represents element-wise multiplication (Hadamard product) [113].

In the LSTM framework, x_t represents the input variable at the present time step, while h_t signifies the output from the preceding cell. Additionally, C_{t-1} denotes the state of the previous cell, providing historical information. These parameters are combined with a series of weight matrices and bias vectors within the logistic sigmoid σ and hyperbolic tangent tanh functions embedded within the input, forget, and output gates. Throughout the learning process, these weights and bias vectors are iteratively refined by comparing the training data.

To achieve accurate predictions, the optimal structure of LSTM networks typically involves the adoption of a single hidden layer [68].

$$(\boldsymbol{n_{in}} + \mathbf{1} + \boldsymbol{n_{ARMA}} + \boldsymbol{n_{WNN}}) \quad \boldsymbol{x} \quad \boldsymbol{n_{hid}} \quad + \quad (\boldsymbol{n_{hid}} \quad + \quad \mathbf{1}) \quad \boldsymbol{x} \quad \boldsymbol{n_{out}} \leq \frac{\mathbf{1}}{\boldsymbol{\alpha}} \boldsymbol{n_{train}} \tag{36}$$

being n_{in} the number of nodes in the input layer, $n_{out}$the number of nodes in the output layer, n_{hid} the number of nodes in the hidden layer, n_{ARMA} is the number of outputs from the ARMA-EGARCH model incorporated into the network, n_{WNN} is is the number of outputs from the Wavelet Neural Network incorporated into the network, n_{train} the number of training data, and α the coefficient ranging from 1 to over 10. To prevent overfitting, α is set higher than 2, considering that the training data double the degrees of freedom in the training process.

4.4. ARMA-EGARCH-Wavelet Neural Networks-Quantum Neural Networks (ARMA-EGARCH-WNN-QNN)

The fusion of CNNs with quantum computing presents a promising computational approach with significant forecasting capabilities [99]. A qubit in quantum computing is a fundamental unit of data, standing for a probabilistic statement. One qubit may occur in states of "1" or "0", or in an arbitrary superposition thereof [50,76]. The status of a qubit reads as described below:

$$|\psi\rangle = \alpha|o\rangle + \beta\ |1\rangle \tag{37}$$

α and β indicate the amplitude of the corresponding states such that $|\alpha|^2 + |\beta|^2 = 1$. It is determined as a pair of numbers $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ The angleθ denotes the specification of geometrical aspects, described as: $\cos(\theta) = |\alpha| and \sin(\theta) = |\beta|$. Using quantum gates may be applied to set likelihoods for weight improvement [112,76]. A rotation gate example is proposed as follows:

$$U(\Delta\theta) = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \tag{38}$$

The quantum gate described above enables the upgrading of a qubit's state. Below, the application of the spin gate on a qubit is detailed.

$$\begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{39}$$

The procedure initiates with a quantum hidden neuron from the state $|o\rangle$, which arranges for the overlay as follows:

$\sqrt{p}|O\rangle + \sqrt{1}\ -p|1\rangle with\ O \leq |p| \leq 1$ (40) p is the random probability of starting the system in the state $|O\rangle$. Classical neurons are induced by generation of random numbers [112,76]. Quantum neuron outputs is expressed as follows:

$$v_j = f\left(\sum_{i=1}^{n} w_{ji} \quad * \quad x_i\right) \tag{41}$$

f shows a problem-dependent sigmoid or Gaussian function. The output from the network is expressed as:

$$y_k = f\left(\sum_{j=1}^{l} w_{jk} \quad * \quad v_j\right) \tag{42}$$

The intended production is the o_k and the upgrading of output layer weight is given in the following formulas:

$$E_k^2 = \frac{1}{2} \quad |y_k - o_k|^2 \tag{43}$$

$$\Delta w_{jk} = \eta e_k f\acute{v}_j \tag{44}$$

The representation of ARMA-EGARCH-WNN output in a Quantum Neural Network (QNN) involves mapping the classical outputs from the ARMA-EGARCH-WNN model into the input layer of the QNN. This process uses specific weight parameters to connect the

outputs of the ARMA-EGARCH-WNN model and bias terms associated with the QNN input. This introduction of ARMA-EGARCH-WNN Output into QNN is defined as [76]:

$$x_i = \sum_{j=1}^{m} a_{ij} \times \omega_j + b_i \tag{45}$$

where a_{ij} denotes the weight parameter responsible for connecting the output of the ARMA-EGARCH-WNN model to the input layer of the QNN. The variable w_j represents the output generated by the ARMA-EGARCH-WNN model, capturing its predictive insights. The term b_i signifies the bias term associated with the input layer of the QNN, contributing to the overall activation of the neural network [4].

Quantum encoding of ARMA-EGARCH-WNN output involves translating the classical predictive results from the ARMA-EGARCH-WNN model into quantum states that can be processed by a Quantum Neural Network (QNN). This encoding transforms the model's output into quantum representations, enabling the integration of advanced quantum computational capabilities with the sophisticated time-series analysis provided by the ARMA-EGARCH-WNN model. This representation is defined by the Eq. (46) [76].

$$\left|\psi\right\rangle = \sum_{j=1}^{m} a_{ij}\left|\omega_j\right\rangle + \left|b_i\right\rangle \tag{46}$$

$|\psi_i\rangle$ represents the quantum-encoded input derived from the ARMA-EGARCH-WNN model. The term $|w_j\rangle$ denotes the quantum state that corresponds to the output of the ARMA-EGARCH-WNN model, encapsulating the predictive information in a quantum format. Additionally, $|b_i\rangle$ signifies the quantum state corresponding to the bias associated with the input of the QNN. These quantum states collectively enable the translation of classical predictive outputs from the ARMA-EGARCH-WNN model into the quantum domain, facilitating advanced quantum computational processing within the QNN framework.

4.5. ARMA-EGARCH-Wavelet Neural Networks-Quantum Recurrent Neural Network (ARMA-EGARCH-WNN-QRNN)

In quantum computing, a system comprising n qubits occupies the n-fold Hilbert space of the tensor product $\mathscr{H} = \left(\mathbb{C}^2\right)^{\otimes d}$, leading to a size of 2^d. A quantum state is shown as a unit vector $\psi \in \mathscr{H}$, commonly despicted in bracket notation $|\psi\rangle \in \mathscr{H}$ |; its conjugate transpose with $\langle\psi| = |\psi\rangle^{\dagger}$; then the inner product $\langle\psi|\psi\rangle = \|\psi\|_2^2$ means the square of the 2-norm of ψ. $[\psi\rangle\langle\psi]$ then calls the outer product, yielding a tensor of rank 2. Ground conditions in computational terms relate to $|0\rangle = (1, 0)$, $|1\rangle = (0, 1)$, while compound ground states, such as $|01\rangle = |0\rangle\bigotimes|1\rangle = (0, 1, 0, 0)$ can be established [17].

A quantum gate is a unitary operation $\mathbb{U}$ applied to a subset $\mathbb{S} \subseteq [n]$ of qubits within a Hilbert space $\mathscr{H}$ The operation $\mathbb{U}\epsilon\mathbb{S}\mathbb{U}\left(2^{|\mathbb{S}|}\right)$ and $\mathscr{H}$ is extended to act as the identity on the remaining qubits, represented as $\mathbb{U}_{\mathbb{S}}\bigotimes 1_{[n]\setminus\mathbb{S}}$. This extension is often not explicitly stated. In a quantum circuit, the first gate, R(θ), performs a unitary operation on a specific qubit, typically the second qubit from the bottom, and depends on the parameter θ. The dashed line on the gate indicates a "controlled" transaction, such as a unitary block-diagonal map, $|0\rangle\langle 0|\bigotimes 1 + |1\rangle\langle 1|\bigotimes\mathbb{R}(\theta) = 1 \oplus \mathbb{R}(\theta)$, which translates to "apply $|1\rangle$ apply $\mathbb{R}(\theta)$." Sequences of gates are computed as matrix products, with detailed circuit descriptions provided by Liu et al. [73].

Projective measures of a single qubit are determined by a hermitian 2× 2 matrix P, such as $M|1\rangle\langle 1| = \mathrm{diag}(0, 1)$; the complementary outcome is then $M^{\perp} = 1 - M$. The measures are conducted within the circuit and yield outcomes in metres. Given a quantum state $|\psi\rangle$, the post-measurement state is $M|\psi\rangle/p$ /p with probability $p = \|M|\psi\|_2$. Such a likelihood is also used as a postselection probability to ensure a balanced outcome MM. To increase this likelihood close to 1, approximately $\sqrt{1/p}$ rounds of amplitude amplification may be applied [51].

Quantum circuits for Variational Quantum Eigensolver (VQE) applications are notably concise, featuring a blend of single-qubit parameterised gates and entangled gates like controlled-not operations. Consequently, they offer the advantage of accommodating a multitude of parameters within a compact circuit design. Despite being recognised as a universal family of circuits, their dense entanglement gate configuration and lack of correlation between parameters contribute to their over-parameterisation. This makes them challenging to train effectively, especially for sorting tasks involving inputs exceeding a few bits [18].

We have developed a meticulously structured parameterised quantum circuit, characterised by the reuse of a small number of parameters across multiple iterations. This circuit is primarily centred around a novel quantum neuron design, which adjusts its target state based on a non-linear activation function linked to the polynomials of its binary inputs. The neuron comprises an input stage that updates the cell's state with each iteration, followed by multiple computational steps to process the input and cell state. Finally, an output step generates a probability density for potential predictions. Utilising these Quantum Recurrent Neural Network (QRNN) cells iteratively over input sequences mirrors the operation of traditional Recurrent Neural Networks (RNNs) [52].

During training, we apply quantum amplitude amplification [52] to the output pathways to accurately measure the relevant tokens from the training data at each step. Although measurements are generally non-unitary operations, the use of amplitude amplification helps ensure that these measurements remain as close to unitary as needed.

Classical neural networks derive their strength from the incorporation of non-linear activation functions applied to the transformations within network layers. In contrast, due to the principles of quantum mechanics, any quantum circuit inherently performs linear operations. However, non-linear behaviour is not absent in quantum mechanics; for instance, consider a single-qubit gate R(θ)= exp(iYθ) for the Pauli matrix Y [83], which acts as a transformative element.

$$R(\theta) = \exp\left(i\theta\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\right) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \tag{47}$$

consider a rotation akin to manoeuvring within the two-dimensional space defined by the computational basis vectors of a single qubit, $\{|0\rangle, |1\rangle\}$. Despite the linearity of the rotation matrix itself, we observe that the state amplitudes — $\cos\theta$ and $\sin\theta$ — vary non-linearly with the angle θ. Extending this rotation to a controlled operation cR(i, θ_i) contingent upon the i^{th} qubit of a state $|x\rangle$ for $x \in \{0,1\}^n$, yields the mapping:

$$R(\theta_0)cR(1,\theta_1 \quad)\ldots cR(n,\theta_n \quad)|x\rangle|0\rangle = |x\rangle(\cos(\eta)|0\rangle + \sin(\eta)|1\rangle)$$

$$\text{for}\eta = \theta_0 + \sum_{i=1}^{n}\theta_i x_i \tag{48}$$

The conversion of cosine amplitudes through a controlled operation is inherently non-linear. However, the sine function lacks a sufficiently distinct "flat" region, similar to that of a linear rectified unit. Cao et al. [22] proposed a method to incorporate a linear mapping onto a set of qubits, resulting in amplitudes displaying steeper inclines and plateaus, resembling a sigmoidal activation function. This activation function is governed by a parameter ord ≥ 1, dictating the slope of the resulting activation amplitudes. The quantum neuron in its pure states undergoes a rotation by an angle $f(\theta) = \arctan\left(\tan(\theta)^{2ord}\right)$, where ord ≥ 1 represents the neuron's order. Assuming an affine transformation η for the input bitstring x_i depicted in Eq. (60), this rotation translates into the amplitudes.

$$\cos(f(\eta)) = \frac{1}{\sqrt{1+\tan(\eta)^{2x2^{ord}}}}\text{and}\sin(f(\eta)) = \frac{\tan(\eta)^{2^{ord}}}{\sqrt{1+\tan(\eta)^{2x2^{ord}}}} \tag{49}$$

that emerges by standardising the transform $|0\rangle \mapsto\!\!\!\rightarrow \cos(\theta)^{2^{ord}}|0\rangle + \sin(\theta)^{2^{ord}}|1\rangle$ as can be easily observed. When ord equals 1, the diagram appears on the left; for ord equals 2, it appears on the right.

The quantum neuron, referred to as a repetition-to-success (RUS) circuit, serves as a key component here. It signals the success of the performed circuit by detecting the measured ring. A measurement yielding zero signifies successful execution of the neuron. Conversely, when the result is one, a correction circuit restores the state to its initial setup. By starting with a pure state (such as$|x\rangle$ for $x \in \{0,1\}^2$ and repeating the process whenever a measurement yields 1, one can achieve a high probability of success.

When dealing with control in superposition, like the state $|x\rangle + |y\rangle/\sqrt{2}$, it doesn't function as expected when x and y represent different bit-strings of length n. The success of this approach hinges on the amplitudes overlapping correctly. A method known as fixed-point oblique amplitude amplification [95] tackles this issue by selectively post-processing measurement results of 0 while ensuring the operation's unitarity with high precision. However, employing this technique requires multiple iterations of quantum circuits, the quantity of which is contingent upon the likelihood of obtaining a measurement result of 0 initially. This likelihood depends on the parameters of the neuron, specifically θ, and the input state provided. It's crucial to note that by opting for sufficiently high individual post-selection probabilities, the overall probability of success across multiple quantum neurons doesn't diminish exponentially.

We enhance the capabilities of the quantum neuron by expanding the range of check terms. Specifically, η as described in Eq. (61), represents an affine transformation of the boolean vector $x = \{x_{1,\ldots,}x_n\}$ for $x_i \in \{0, 1\}$. By introducing multi-control gates with individual parameterized rotations, denoted by a multi-index θ_I which varies depending on the qubits i ∈ I on which the gate conditions are applied, we open up the possibility of incorporating higher-degree polynomials.

$$\eta' = \theta_0 + \sum_{i=1}^{n}\theta_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n}\theta_{ij}x_i x_j + \ldots = \sum_{\substack{I\subseteq[n] \\ |I|\le d}}\theta_I\prod_{i\in I}x_i \tag{50}$$

We consider the degree of the neuron, denoted as d. When d equals 2 and n equals 4, we can demonstrate how a checked rotation increases to a higher-order transformation η' on the bit string x_i. This means that higher-degree boolean logic operations can be embedded directly within a single conditional rotation, i.e, an AND operation between two bits x_1 and x_2 is x_1x_2. By integrating aspects of ARMA, EGARCH, and WNN methodologies into the QRNN framework, the Eq. (51) endeavors to capture intricate dependencies and patterns within time series data. It enables the incorporation of historical data, autoregressive components (ARMA), conditional volatility (EGARCH), and additional patterns identified by the Wavelet Neural Network (WNN) [95].

$$\eta' = \theta_0 + \sum_{i=1}^{n}\theta_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n}\theta_{ij}x_i x_j + \sum_{k=1}^{m}\theta_k x_{AR_k} + \sum_{l=1}^{p}\theta_i x_{EG_l} + \sum_{q=1}^{r}\theta_q x_{WNN_q} \tag{51}$$

being the variable η′ represents the transformed output, which could denote the result of a neuron's operation or the concealed state within a neural network layer. θ_0 acts as the bias term or intercept, denoting the baseline value of the transformed output when all other input features are absent. θ_i symbolizes the coefficients or weights assigned to individual input features x_i, delineating their respective influences on the transformed output. x_i represents the input features, encompassing various components such as historical data, ARMA and EGARCH terms, and potentially WNN terms, each elucidating different aspects of the input data. Coefficients θij

correspond to the interaction between input features x_i and x_j, including the combined effect of two features on the transformed output. x_{ARk} embodies the ARMA terms, signifying the k-th autoregressive term reflecting lagged values of the time series data. x_{EGl} signifies the EGARCH terms, specifically the llth conditional volatility term indicative of conditional volatility in the time series data. Lastly, x_{WNNq} denotes the WNN terms, specifically the qqth term derived from the Wavelet Neural Network, capturing additional patterns or features in the time series data identified by the WNN. To put the QRNN cell into action, we need to apply it iteratively to a sequence of input words, denoted as$in_1, in_2, \ldots, in_L$. The outgoing lanes out_i represent a discrete distribution p_i over the class labels [95].

4.6. Hyperparameter settings for each algorithm

In the following sections, we provide a detailed overview of the hyperparameters for each algorithm. The corresponding values are organized and presented in Tables 1 to 5 for clarity and reference.

5. Results

The Results section presents the findings of our comprehensive analysis of the potential of Bitcoin to trigger speculative pressures on the US Dollar. Through various Machine Learning models, including ARMA-EGARCH-WNN variations, we examined the behavior of the Bitcoin/US dollar and its influence on US dollar/EUR, through an agent-based simulation for the experimental validation as the purpose, in the contexts of speculative attacks and short squeezes. This section details the performance metrics, including Precision, Recall, F1 Score, False Positive Rate, False Negative Rate, and Mean Squared Error (MSE), across different models, highlighting their effectiveness in capturing market dynamics. The results offer valuable insights into the interconnectedness of cryptocurrency and fiat currency markets, elucidating the implications of Bitcoin's volatility on broader financial stability.

5.1. Data

We have taken the data for Bitcoin/US dollar and US dollar/EUR markets from Yahoo Finance (https://finance.yahoo.com/) and CoinMarketCap (https://coinmarketcap.com). The samples used to train and test the models are from 17 September 2014–4 June 2024. According to the models and thresholds used in Section 3, we have registered 44 attempts of speculative attacks and 61 attempts of short squeezes during that period. Python version 3.10 was employed for the estimation of ARMA-EGARCH-WNN-NN models. Key libraries included stats models for ARMA modeling, arch for EGARCH estimation, and tensorflow for implementing the Wavelet and NN components, ensuring seamless execution of these advanced techniques.

5.2. Wavelet coherence

Wavelet coherence is a useful tool for analyzing the dynamic relationship between two-time series across different frequencies and time scales. In this context, the analysis of wavelet coherence between Bitcoin/US dollar and US dollar/EUR can provide insights into how movements in the cryptocurrency market, specifically Bitcoin, influence traditional forex markets, or vice versa. If the coherence between Bitcoin/US dollar and US dollar/EUR is high at certain frequencies, this indicates a strong correlation between Bitcoin's price movements and the US dollar to EUR exchange rates during those time intervals. This suggests that certain factors affecting Bitcoin are also impacting the U.S. dollar in the traditional Forex market. For example, high wavelet coherence might indicate that movements in Bitcoin's price relative to the dollar are transmitting influences to the forex market, thus affecting the US dollar/EUR pair. This could happen during periods of high volatility in the cryptocurrency market that impact the perception of the US dollar as a safe or risky asset

Table 1
Hyperparameters values for ARMA-EGARCH-WNN-RNN.

Hyperparameter	Description	Value
p	ARMA AR Part Order	3
q	ARMA MA Part Order	2
GARCH Order	GARCH terms order in EGARCH	1
ARCH Order	ARCH terms order in EGARCH	1
Wavelet Function	Type of wavelet function used in WNN	Morlet
Number of Wavelet Levels	Number of decomposition levels in WNN	3
Number of Hidden Layers	Number of hidden layers in RNN	2
Number of Neurons per Layer	Number of neurons in each hidden layer of RNN	100
Learning Rate	Learning rate for training the RNN	0.005
Batch Size	Number of samples per batch during training	64
Epochs	Number of training iterations	200
Activation Function (σ)	Activation function for hidden layers	tanh
Activation Function (o)	Activation function for output layer	sigmoid
Optimizer	Optimization algorithm used for training	Adam, SGD
Dropout Rate	Dropout rate for regularization	0.3
Sequence Length	Length of input sequences for the RNN	30
Regularization	Regularization parameter for RNN	L2 (0.001)

Table 2
Hyperparameters values for ARMA-EGARCH-WNN-CNN.

Hyperparameter	Description	Value
p	ARMA AR Part Order	3
q	ARMA MA Part Order	2
GARCH Order	GARCH terms order in EGARCH	1
ARCH Order	ARCH terms order in EGARCH	1
Wavelet Function	Type of wavelet function used in WNN	Morlet
Number of Wavelet Levels	Number of decomposition levels in WNN	3
Number of Convolution Layers	Number of convolutional layers in CNN	2
Number of Filters	Number of filters (kernels) in each convolution layer	64
Filter Size	Size of the convolution filters	3×3
Stride	Stride size for the convolution operation	2
Padding	Padding type for the convolution layers	'same'
Pooling Size	Size of the pooling window	2×2
Pooling Type	Type of pooling (max or average)	'max'
Activation Function (ReLU)	Activation function for convolutional layers	ReLU
Learning Rate	Learning rate for training the CNN	0.001
Batch Size	Number of samples per batch during training	128
Epochs	Number of training iterations	200
Optimizer	Optimization algorithm used for training	SGD
Dropout Rate	Dropout rate for regularization	0.3
Regularization	Regularization parameter for CNN	L2 (0.001)

Table 3
Hyperparameters values for ARMA-EGARCH-WNN-LSTM.

Hyperparameter	Description	Value
p	ARMA AR Part Order	3
q	ARMA MA Part Order	2
GARCH Order	GARCH terms order in EGARCH	1
ARCH Order	ARCH terms order in EGARCH	1
Wavelet Function	Type of wavelet function used in WNN	Morlet
Number of Wavelet Levels	Number of decomposition levels in WNN	3
Number of LSTM Layers	Number of LSTM layers	2
Number of LSTM Units	Number of units in each LSTM layer	100
Activation Function	Activation function for LSTM layers	Tanh
Recurrent Activation	Activation function for LSTM gates	sigmoid
Learning Rate	The learning rate for training the LSTM	0.001
Batch Size	Number of samples per batch during training	64
Epochs	Number of training iterations	200
Optimizer	Optimization algorithm used for training	Adam
Dropout Rate	Dropout rate for regularization	0.3
Sequence Length	Length of input sequences for the LSTM	30
Regularization	Regularization parameter for LSTM	L2 (0.001)
α (Alpha)	Coefficient for preventing overfitting	5

[36].

The arrows in the coherence graph indicate the phase difference between the two series. Arrows pointing in the same direction suggest that the series are in phase, meaning they rise or fall together. Conversely, opposite arrows indicate they are out of phase. Understanding these phases is crucial to identifying whether Bitcoin/US dollar is leading or following US dollar/EUR in terms of price movements, which can provide valuable information for trading strategies.

The three charts provide a detailed analysis of the dynamic relationship between the Bitcoin/US dollar and US dollar/EUR currency pairs, using both time representation and wavelet coherence.

In the first two charts (Fig. 1), the time series of Bitcoin/US dollar and US dollar/EUR are shown, respectively, over a period of 2 seconds. In the Bitcoin/US dollar chart, a series of price data shows a notable increase in volatility starting around 0.6 seconds, evidenced by broader and more frequent oscillations. This behavior may reflect specific market events affecting Bitcoin, such as major announcements or significant movements in the cryptocurrency market. In the US dollar/EUR chart, a time series of prices is also observed, but with a broader range of values than Bitcoin/US dollar. Starting around 0.8 seconds, there is an increase in variability, which could be related to events in the forex market affecting the dollar and the euro.

The third chart (Fig. 2) shows the wavelet coherence between Bitcoin/US dollar and US dollar/EUR. This frequency-time domain visualization uses the axes to represent time (in seconds) and period (also in seconds), while the color scale indicates the magnitude of coherence. Yellow areas indicate high coherence, suggesting that the Bitcoin/US dollar and US dollar/EUR price series are strongly correlated at those specific frequencies and times. A notable area of high coherence is found in the periods from 0.02 to 0.1 seconds and between 0.5 and 1.2 seconds, implying synchronization of price movements at these scales. The arrows within these areas indicate the

Table 4
Hyperparameters values for ARMA-EGARCH-WNN-QNN.

Hyperparameter	Description	Value
p	ARMA AR Part Order	3
q	ARMA MA Part Order	2
GARCH Order	GARCH terms order in EGARCH	1
ARCH Order	ARCH terms order in EGARCH	1
Wavelet Function	Type of wavelet function used in WNN	Morlet
Number of Wavelet Levels	Number of decomposition levels in WNN	3
Quantum Layers	Number of layers in the Quantum Neural Network	2
Qubits per Layer	Number of qubits in each layer of the Quantum Neural Network	4
Quantum Gate	Type of quantum gate used in the Quantum Neural Network	Rotation
Learning Rate	Learning rate for training the Quantum Neural Network	0.001
Batch Size	Number of samples per batch during training	64
Epochs	Number of training iterations	200
Optimizer	Optimization algorithm used for training	Adam
Dropout Rate	Dropout rate for regularization	0.3
Sequence Length	Length of input sequences for the Quantum Neural Network	30
Regularization	Regularization parameter for Quantum Neural Network	L2 (0.001)
Quantum Superposition	Probability distribution for initializing quantum states	0.5
Activation Function	Activation function for quantum neurons	Sigmoid
Weight Initialization	Method for initializing weights in the Quantum Neural Network	Random
Entanglement Layers	Number of entanglement layers in the Quantum Neural Network	2
α (Alpha)	Coefficient for preventing overfitting	5

Table 5
Hyperparameters values for ARMA-EGARCH-WNN-QRNN.

Hyperparameter	Description	Value
ARMA Order (p)	ARMA AR Part Order	3
ARMA Order (q)	ARMA MA Part Order	2
GARCH Order	GARCH terms orders in the EGARCH model	1
ARCH Order	ARCH terms orders in the EGARCH model	1
Wavelet Function	Type of wavelet function used in WNN	Morlet
Number of Wavelet Levels	Number of decomposition levels in WNN	3
Quantum Layers	Number of layers in the Quantum Recurrent Neural Network	2
Qubits per Layer	Number of qubits in each layer of the QRNN	4
Quantum Gate	The type of quantum gate used in the QRNN	Rotation
Learning Rate	The learning rate for training the QRNN	0.001
Batch Size	Number of samples per batch during training	64
Epochs	Number of training iterations	200
Optimizer	Optimization algorithm used for training	Adam
Dropout Rate	Dropout rate for regularization	0.3
Sequence Length	Length of input sequences for the QRNN	30
Regularization	Regularization parameter for the QRNN	L2 (0.001)
Quantum Superposition	Probability distribution for initializing quantum states	0.5
Activation Function	Activation function for quantum neurons	Sigmoid
Weight Initialization	Method for initializing weights in the QRNN	Random
Entanglement Layers	Number of entanglement layers in the QRNN	2
α (Alpha)	Coefficient for preventing overfitting	5

relative phase between the two data series; arrows pointing in the same direction imply in-phase movements (both currency pairs rise or fall together), while opposite arrows indicate out-of-phase movements.

Overall, the analysis of these charts suggests a dynamic interrelation between Bitcoin/US dollar and US dollar/EUR, particularly at certain time scales. High coherence at specific periods indicates that events affecting Bitcoin may have a correlative impact on the forex market, particularly on the US dollar/EUR pair, or vice versa. Traders and investors can leverage this relationship to predict market movements or for hedging strategies. Additionally, understanding the phase difference between the time series can provide deeper insights into which market may be leading or influencing the other in terms of price behavior.

5.3. Speculative attacks and short squeezes prediction with real-time data

The evaluation of various forecasting models—ARMA-EGARCH-WNN-RNN, ARMA-EGARCH-WNN-CNN, ARMA-EGARCH-WNN-LSTM, ARMA-EGARCH-WNN-QNN, and ARMA-EGARCH-WNN-QRNN—for predicting speculative attacks and short squeezes reveals significant differences in performance (Table 6). In terms of training accuracy, the ARMA-EGARCH-WNN-QNN model demonstrated the highest accuracy for speculative attacks at 92.93 %, indicating its superior capability in learning from training data. The ARMA-EGARCH-WNN-LSTM model, with an accuracy of 90.34 %, followed, while the ARMA-EGARCH-WNN-RNN model had the lowest

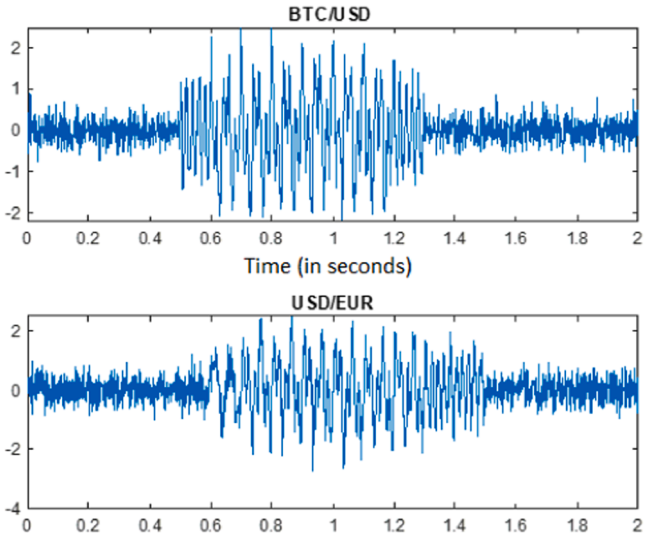

Fig. 1. Time Series Analysis of Bitcoin/US dollar and US dollar/EUR Price Movements.

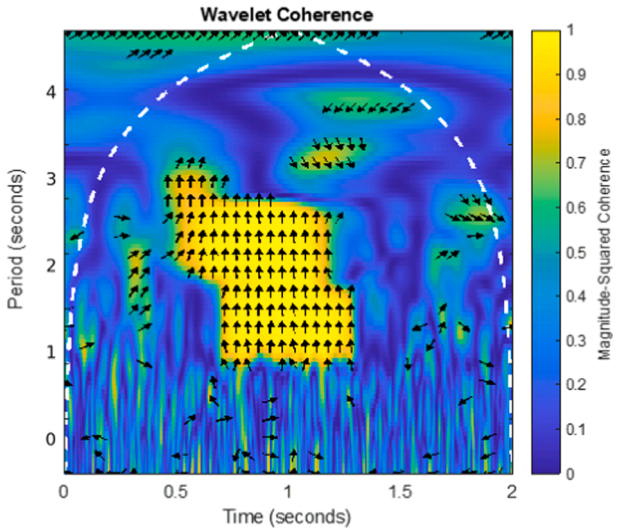

Fig. 2. Wavelet Coherence between Bitcoin and USD Dollar.

accuracy at 90.21 %. This suggests that although RNNs contribute to the learning process, they may not be as effective in capturing the complexities of speculative attacks as QNNs. For short squeezes, the ARMA-EGARCH-WNN-QRNN model excelled with a training accuracy of 94.85 %, highlighting its effectiveness in capturing intricate dynamics during training. The ARMA-EGARCH-WNN-LSTM model, achieving a training accuracy of 92.20 %, also performed well but lagged behind the QRNN, while the ARMA-EGARCH-WNN-CNN model had the lowest accuracy at 92.16 %, suggesting that CNNs may be less suited for capturing the temporal dependencies required for short squeezes.

In the testing phase, the ARMA-EGARCH-WNN-QRNN model continued to demonstrate superior performance, achieving the highest accuracy of 91.23 % for speculative attacks and 91.35 % for short squeezes. This consistency underscores its robustness and reliability in generalizing to new data. Conversely, the ARMA-EGARCH-WNN-RNN model had the lowest testing accuracy for speculative attacks at 88.75 %, indicating less effectiveness in real-world applications. For short squeezes, the ARMA-EGARCH-WNN-CNN model, with the lowest testing accuracy of 89.13 %, showed reduced effectiveness in prediction.

Regarding root mean square error (RMSE) (Table 6), the ARMA-EGARCH-WNN-QRNN model achieved the lowest training RMSE for speculative attacks at 0.0396 and for short squeezes at 0.0381, reflecting its ability to minimize training errors effectively. In contrast, the ARMA-EGARCH-WNN-LSTM model had the highest training RMSE for speculative attacks at 0.0513. For testing, the ARMA-EGARCH-WNN-QRNN model also exhibited the lowest RMSE of 0.0607 for speculative attacks and 0.0492 for short squeezes, demonstrating its superior accuracy and minimal error in generalizing to new data. The ARMA-EGARCH-WNN-LSTM model had the

Table 6
Performance Comparison of ARMA-EGARCH-WNN Models for Forecasting Speculative Attacks and Short Squeezes.

	ARMA-EGARCH-WNN-RNN	ARMA-EGARCH-WNN-CNN	ARMA-EGARCH-WNN-LSTM	ARMA-EGARCH-WNN-QNN	ARMA-EGARCH-WNN-QRNN
Training					
Speculative Attacks	90.210	90.278	90.337	91.623	92.928
Short Squeezes	92.185	92.156	92.204	93.517	94.849
Testing					
Speculative Attacks	88.747	88.561	88.688	89.951	91.232
Short Squeezes	89.155	89.127	88.805	90.069	91.352
RMSE Training					
Speculative Attacks	0.043	0.046	0.051	0.045	0.040
Short Squeezes	0.040	0.043	0.049	0.043	0.038
RMSE Testing					
Speculative Attacks	0.066	0.071	0.079	0.069	0.061
Short Squeezes	0.059	0.072	0.063	0.056	0.049

highest testing RMSE for speculative attacks at 0.0787, indicating less effective performance. The ARMA-EGARCH-WNN-CNN model had the highest testing RMSE for short squeezes at 0.0712, reflecting its less effective generalization capability.

Overall, the ARMA-EGARCH-WNN-QRNN model consistently outperformed the others in both training and testing phases, showcasing its robustness and effectiveness in forecasting speculative attacks and short squeezes. Its superior accuracy and lower RMSE values highlight its capability to capture and generalize the underlying patterns of financial market phenomena effectively. While other models like ARMA-EGARCH-WNN-LSTM and ARMA-EGARCH-WNN-RNN show strong performance, they fall short of the QRNN model's capabilities, particularly in real-world testing scenarios. This indicates that integrating quantile regression with neural networks, as seen in the QRNN model, offers a highly effective approach for accurate and reliable forecasting.

The learning curves presented in the Fig. 3 illustrate the training and validation loss over 500 epochs for various ARMA-EGARCH-WNN models, including RNN, CNN, LSTM, QNN, and QRNN architectures. Each subplot compares the performance for speculative attacks and short squeezes, providing insights into how these models adapt and generalize over time.

In general, all models show a consistent decrease in both training and validation losses, indicating that they are effectively learning the underlying patterns in the data. This trend is an essential indicator of the models' ability to generalize from the training data to unseen validation data, which is crucial for predicting financial time series. Specifically, the QRNN models demonstrate the lowest final training and validation losses among all the architectures. This suggests that QRNN models might be more effective at capturing complex dependencies and nonlinearities in the financial data, leading to better performance in predicting speculative attacks and short squeezes.

The smooth decline in losses with minimal fluctuations across epochs for all models reflects a stable and robust training process. It indicates that the models are not only learning effectively but also avoiding overfitting, as evidenced by the close alignment of training and validation loss curves. Overfitting would typically manifest as a significant divergence between training and validation losses, which is not observed here. Moreover, the slight differences in the performance of various models highlight the importance of model selection in financial time series forecasting. While all models show good learning behavior, the QRNN's superior performance suggests it might be the preferred choice for tasks requiring high accuracy and robustness in predicting market behaviors like speculative attacks and short squeezes.

The ARMA-EGARCH-WNN-QRNN model stands out for its superior accuracy and lower RMSE values, aligning well with recent research that advocates for the combination of advanced techniques like quantile regression and neural networks. This result is consistent with studies by Huang and Wang (2018) and Wang, Sarker, and Bouri [101], which highlight the effectiveness of hybrid models integrating wavelet transformations with neural networks to capture complex, multi-scale features of financial time series. The robust performance of the QRNN model in managing Bitcoin's inherent volatility and non-linear characteristics supports the literature's endorsement of such advanced hybrid approaches.

The use of wavelet transformation to decompose Bitcoin price data aligns with the findings of Dyhrberg [40] and Lee, Koh, and Choe [70], who emphasize the advantages of wavelet-based methods for enhancing forecasting accuracy. By providing a multi-scale view, wavelet transformations allow the ARMA-EGARCH-WNN-QRNN model to capture both short-term fluctuations and long-term trends more effectively. This supports the literature's assertion that wavelet techniques improve the handling of financial data's inherent volatility and non-linearity.

In contrast, the ARMA-EGARCH-WNN-RNN and ARMA-EGARCH-WNN-CNN models performed less favorably, especially in the testing phases. This outcome diverges from the work of Graves (2013) and [69], who highlight the strengths of RNNs and CNNs in sequential data and feature extraction. The underperformance of these models in handling wavelet-transformed data suggests limitations in their ability to manage the multi-scale and temporal aspects critical for accurate financial forecasting. This discrepancy implies that while RNNs and CNNs are valuable in many contexts, they may require further adaptation to effectively process complex financial data.

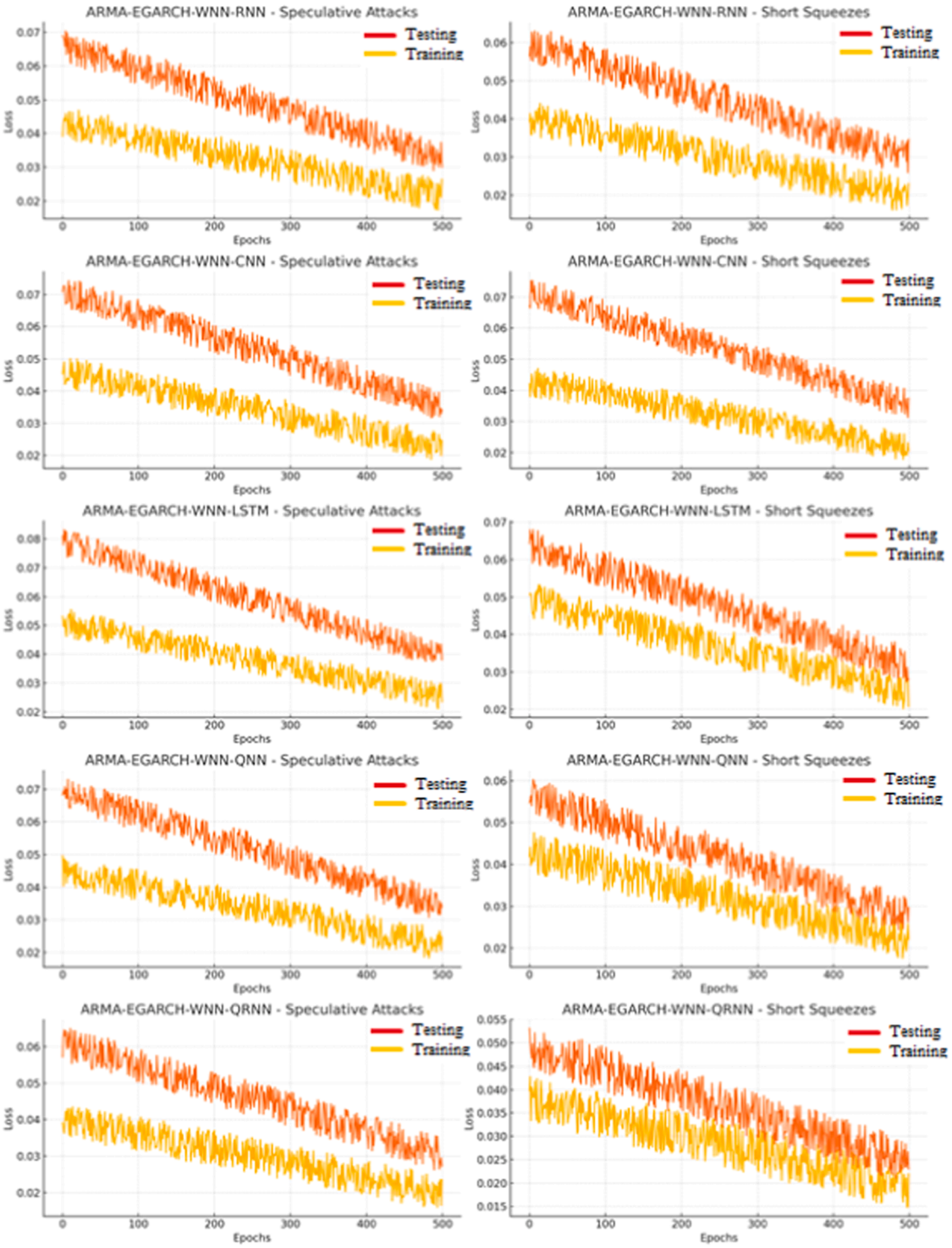

Fig. 3. Learning Curves for ARMA-EGARCH-WNN Models for Speculative Attacks and Short Squeezes.

The ARMA-EGARCH-WNN-LSTM model exhibited strong training performance but struggled in generalization during testing. This observation aligns with Hochreiter and Schmidhuber's [55] findings that LSTMs excel at capturing long-term dependencies but may face challenges in generalizing with complex, multi-scale data. This result supports Wu [103], who notes that LSTMs might not fully capture dynamic interactions in high-frequency financial data, reflecting the need for advanced modeling techniques.

5.4. Experimental validation

Genetic Algorithms (GAs) are usually employed to find optimal solutions to problems by starting with a starting population of solutions and using random processes to create new ones. Over time, these solutions are refined to approach the best possible outcome. GAs are regarded as a simple method for tackling complex non-linear issues, as they are based on concepts such as “reproduction” and “breeding” in a starting population. Key features like “selection, crossover, and mutation” help to generate more efficient solutions, as noted by Liu et al. [74].

The population size in a GA is crucial, determining the total solutions and significantly impacting the outcomes, as highlighted by Moayedi et al. [80]. The term "generations" refers to the iterations in the optimization process. In practice, GAs have been shown to offer significant advantages in optimizing resource allocation to enhance cost-effectiveness and output quality, as discussed by Liu et al. [74].

In our research, we use the GA deployment described by Boboc and Dinică [21], specifically shaped to represent a trading situation. In this GA, subjects are embodied by a collection of rules of technical analysis, known as chromosomes, each containing two or more parameters, referred to as genes.

The algorithm follows these steps:

1. The GA begins by generating an initial population of chromosomes through random selection. The algorithm creates 100 individuals, each representing a set of technical analysis rules with randomly assigned parameters.
2. During the training phase, each subject is assessed according to the profit or loss generated by the trading strategies associated with them. This financial outcome serves as a measure of their performance.
3. The GA then ranks the 100 subjects in terms of their profit or loss, using this as a fitness function. This ranking helps establish the comparative fitness of each individual within the population, guiding the selection process for subsequent generations.
4. The GA addresses to create the next generation through the following steps:

a. Elitism: The most profitable subject from the current generation is retained and included in the new generation.
b. Parent Selection: Each of the 100 subjects has a chance to become a parent for the next generation based on their profit, with higher profits increasing the likelihood. Individuals are categorized into ten groups, with selection probabilities assigned according to their top profit rankings.
c. Offspring Generation: Using their assigned probabilities, pairs of individuals are randomly selected as parents. Following the method of Boboc and Dinică [21], each parent has 24 genes numbered 1–24. A random number "n" is chosen between 1 and 24, and the new offspring inherits genes 1 to "n" from one parent and genes "n + 1" to 24 from the other parent, resulting in 80 new individuals.
d. Diversity Promotion: To further enhance diversity, 19 additional individuals are generated randomly. This ensures a varied population, fostering a broad exploration of potential solutions.

1. Once the GA has created a new generation of 100 individuals, this population replaces the previous one. The GA then repeats the process of evaluating, sorting by profit, and generating the next population. Such an iterative cycle goes on throughout as the AG progressively narrows the population to find optimum solutions.
2. The GA keeps iterating through stages 2–5 until it achieves 80 iterations. Research indicates that adding the number of iterations after this point has no significant effect on the results. Thus, 80 iterations are considered sufficient for achieving reliable outcomes.

The Metropolis-Hastings (MH) algorithm is a well-known and advanced sampling technique (Heratha and Herath, 2018). It operates based on Markov chains and resembles rejection sampling methods, as it only requires proposing a test value without needing to know the normalization of the distribution being sampled. The algorithm draws inspiration from the behavior of systems in statistical mechanics near equilibrium. The system’s evolution is governed by the probability of transitioning from state X to state Y. To reach equilibrium, the system must have an equal probability of being in state X and transitioning to state Y, as expressed by the detailed balance equation,

$$f(X)P(Y|X) = f(Y)P(X|Y) \tag{52}$$

in which $f(X)P(Y|X)$ denotes the expectation $f(X)$ for meeting the system in the proximity of X times the conditional probability $P(Y|X)$ that the system recovers from X to Y (Ayekple, Tetteh and Fefemwole, 2018). Given a physical procedure, $P(Y|X)$ is known, and one tries to find $f(X)$. The MH algorithm attempts the opposite: for $f(X)$, one tries to find the transition probability that leads to an equilibrium system. Transitions from X to Y are suggested according to any test distribution $T(Y|X)$. It compares $f(Y)$ with $f(X)$ and it accepts Y with likelihood $A(Y|X)$. Hence,

$$P(Y|X) = A(Y|X)\ T(Y|X) \tag{53}$$

We construct a Markov chain consisting of the states $X_0, X_1, X_2, \ldots X_N$ that are accessed at every transition, starting from X_0 (Heratha and Herath, 2018). Each X_n becomes a random variable to fulfill the next equation (Ayekple, Tetteh and Fefemwole, 2018).

$$\lim_{n\to\infty} \varnothing_n(X) = f(X) \tag{54}$$

A transition probability *T(Y|X)* exists at each step of the random walk, and this is standardised,

$$\int dYT(Y|X) = 1 \tag{55}$$

On the assumption from Y to X is always possible if it is possible to go from X to Y, or vice versa, the definition is:

$$q(Y|X) = \frac{T(X|Y)f(Y)}{T(X|Y)f(X)} \geq 0 \tag{56}$$

From this one on *q(Y|X)* the probability of acceptance can be,

$$A(Y|X) = mn\ \{q(Y|X),\ 1\} \tag{57}$$

5.4.1. Application domain with problem

This paper's model extends the findings from the investigation led by Cocco and Marchesi [33]. It integrates practical trading strategies to emulate Bitcoin/US dollar trading and encapsulate the significant traits evident in bitcoin price trends. The method employed in this work is agent-based, a methodology extensively acknowledged in relevant literature [27,28,29] for its efficacy. Our aim is to estimate speculative attacks and short squeezes in the Bitcoin/USD market.

Additionally, Gilli and Winker [48] employ computational techniques to launch the Kirman [63,64] model, linking herding behaviour based on distinct reactions to the actions of participants in the FOREX market. According to Barde [12], the model comprises N agents divided into fundamentalists (f) and chartists (c). Switching probabilities regulate the dynamics of the framework:

$$\begin{cases} P_t^{c\to f} = (1 - x_t)(\varepsilon + \rho x_t), \\ P_t^{f\to c} = x_t(\epsilon + \rho(1 - x_t)\,), \end{cases} \tag{58}$$

being x_t the population fraction of fundamentalists at time t, where ranges from 1 to T, ε represents the likelihood of a spontaneous switch between f and c groups, and ρ shows the likelihood of a satisfactory engagement. Price prospects p_t of the two groups are subsequently formulated as:

$$\begin{cases} E^f(\Delta p_t) = \phi(\overline{p} - p_{t-1}), \\ E^c(\Delta p_t) = p_{t-1} - p_{t-2,} \end{cases} \tag{59}$$

The adjustment speed to the fundamental price $\overline{p}$ is represented by ϕ.

The baseline parameterization aligns with the initial Gilli and Winker [48] configuration, with N set at 100 agents. Switching probabilities are established at $\rho = 0.264$ and $\varepsilon = 0.0001$, while the adjustment speed is defined as $\phi = 0.0225$. The fundamental price $\overline{p}$ s fixed at 1000.

Initially, our attention is directed towards the probabilities of spontaneous switching, ε, and successful recruitment, ρ, seen as pivotal elements shaping the model's dynamics. The hypothesis posits that augmenting both probabilities amplifies the model's complexity. Additionally, we examine the influence of the adjustment speed ϕ on the fundamental price φ and the total number of individual traders, N, rounded to whole numbers. Generally, augmenting N is anticipated to enhance the model's complexity, as a larger population of agents is likely to manifest more intricate behaviours. The effect of the adjustment speed ϕ is observed without preconceived notions.

This study models an artificial foreign exchange market with two types of participants: "random traders and chartists". Random traders order without a discernible pattern, driven by reasons such as portfolio diversification or liquidity needs, more than speculation. They issue purchase and sale instructions equally likely. Chartists, on the other hand, make trading decisions based on specific technical analysis principles. Their goal is to profit from market movements by opening positions and issuing buy or sell orders based on their trading strategies. After testing a set of trading rules, including both initial and ongoing strategies, chartists aim to time their trades effectively to maximize profits or minimize losses [5].

The study leverages various predictive rules to forecast price movements, inspired by Chiarella et al. [31] and LiCalzi and Pellizzari [72], who used similar rules to predict stock market returns. Chartists use technical indicators like the "Exponential Moving Average", "Moving Average Convergence Divergence", "Relative Strength Index", and "Genetic Filter" to make trading decisions. Additionally, a GA is employed to identify the most profitable trading rules, following a method similar to Tedeschi et al. [96], where traders replicate the strategies of successful traders. For this study, the AG chooses the negotiation standards with the greatest benefits.

At the initial period t = 0, every trader in the market possesses a starting sum $c_i(0)$ of US dollars and an initial quantity $b_i(0)$ of bitcoins, where i represents the trader's index. When a trader enters the market at time $t_i^E > 0$, they only hold a sum c_i (t_i^E) of USA dollars, marked with the superscript E indicating their entry.

To distribute the bitcoins, we randomly assigned them to a portion of traders. This approach aligns with the Gibrat principle of preferential attachment, which suggests that wealthier subjects are more likely to receive larger portions [106,47]. At the start, the wealth distribution among traders, involving both bitcoins and US dollars, conforms to Zipf's law (as explained in the research by [71]). This distribution is established before the simulation, employing the method detailed in the studies conducted by Cocco, Concas, and Marchesi [32] and Cocco and Marchesi [33].

To ascertain the traders, denoted as N_T, active in the market at a specific period t, we employed a fitting curve. This curve, represented as N_T (t), was adjusted applying seven data points, according to Cocco and Marchesi [33].

The number of traders, denoted as N_T, is modelled using a fitting curve that follows a general exponential:

$$N_T(t) = a * \exp^{b*t} \tag{60}$$

The optimal fitting parameters, denoted as a and b, are found to be 1.744e+04 and 0.002465 accordingly. The starting t value is established as 1824, representing January 1st, 2014, making the beginning of our simulations. This choice is consistent with the methodology advocated in the studies conducted by Cocco, Concas, and Marchesi [32] as well as Cocco and Marchesi [33].

Following the method mentioned above, when a trader joins the market at a given time t, they are categorised into either the Chartists' group or the Random Traders' group. The likelihood of being part of the Chartists' group, denoted as percC, is established at 0.6, whereas the likelihood of being part of the Random Traders' group, denoted as percR, is set at 0.4. It's worth noting that the sum of percC and percR value is 1.

In the trading domain, chartists adhere to a defined set of trading regulations, which they randomly select upon entering the market. These rules can either be optimal or random. The likelihood of chartists employing the best rules, denoted as $percC_B$, stands at 0.48, while the probability of employing random rules, denoted as $percC_R$, is 0.12. It's important to note that the sum of $percC_B + percC_R = perC$. The optimal rules are determined through a Genetic Algorithm (GA), which selects rule parameters to maximise operators' profitability. Conversely, the parameters for random rules are chosen randomly without optimisation. Subsequently, the chartists who adopt the best rules are referred to asC_{bR}, while the remaining chartists are known as C_{rR}. Chartists operate by applying their chosen rule settings, selected either from established standards or generated randomly [32,33].

In our model, similar to the study by Boboc and Dinica (2013), we have a total of 6 rules, comprising 4 for initiating a position and 2 for closing a position, resulting in 24 parameters. These details are presented in Table 7 for easy interpretation.

Standards for the position creation are as follows:

Rule 1: Filter, founded on 5 parameters:

1. $filter_{flag}$ enables or disables the rule. A score of 0 disables the rule, and a score of 1 enables the rule.
2. $filter_{periods}$, marked n. Existing price for consideration.
3. $filter_{increaseS}$, written p;
4. $filter_{decreaseS}$, scored q;

   Three and four are variables that show the exchange rate between two currency pairs.
5. $filter_{booleanS}$ means a Boolean symbol indicating trading patterns.

   Rule 2: RSI, as:

Table 7
Model rules and parameters.

Rules	Parameters
Filter	$filter_{flag}$
	$filter_{periods}$
	$filter_{increaseS}$
	$filter_{decreaseS}$
	$filter_{booleanS}$
Relative Strength Index (RSI)	rsi_{flag}
	rsi_{n}
	rsi_{os}
	rsi_{ob}
	$rsi_{boolean}$
Exponential Moving Average (EMA)	ema_{flag}
	ema_{n}
Moving average convergence divergence (MACD)	$macd_{flag}$
	$macd_{periodsS}$
	$macd_{periodsL}$
	$macd_{periodsN}$
	$macd_{booleanS}$
Fixed exit levels (FEL)	fel_{flag}
	fel_{tp}
	fel_{sl}
Trailing exit levels (TEL)	tel_{flag}
	tel_{tp}
	tel_{sl}
	tel_{tl}

$$RSI_t(n) = \frac{\sum_{i=t-n}^{t} \frac{\max_i(P_i - P_{i-1}, 0)}{n}}{\sum_{i=t-n}^{t} \frac{\max_i(P_i - P_{i-1}, 0)}{n} + \sum_{i=t-n}^{t} \frac{\max_i(P_{i-1} - P_i, 0)}{n}} \tag{61}$$

The RSI calculates buy and sell signals based on the closing price of the period (denoted as P) and the number of periods (denoted as n) used in its calculation. These signals are determined by whether the RSI surpasses or falls below the thresholds for overbought and oversold conditions. The RSI is determined according to the following parameters:

6. rsi_{flag} enables or disables the rule.
7. rsi_n refers to the terms to be inserted when calculating the average maxima and minima.
8. rsi_{os};
9. rsi_{ob};

The eight and nine parameters represent the oversold and overbought indicators

10. $rsi_{boolean}$ represents a binary signal that indicates trading signals, with values of either 0 or 1 occurring with equal probability. If takes the value 0, it signals a buy when the RSI value is below rsi_{os}, and a sell when the RSI value exceeds rsi_{ob}.

Rule 3: EMA is a weighted moving average as follows:

$$EMA_t(n, Close) = \frac{2}{n+1} Close_t + EMA_{t-1}\left(1 - \frac{2}{n+1}\right) \tag{62}$$

At the previous moment, EMA, n, and Close respectively represent the value of this indicator, the number of periods, and the closing price of the period.

This rule is formed by two parameters:

11. ema_{flag} enables and disables the rule.
12. ema_n means the nth period.

Rule 4: MACD, defined as:

$$MACD_t(p, q, m) = MACD_t(p, q) - Signal_t(m) \tag{63}$$

being:

$$MACDline_t(p, q) = EMA_t(p, close) - EMA_t(q, close) \tag{64}$$

$$SignalLine_t(m) = EMA_t(m, \quad MACD_t(p, q)\,) \tag{65}$$

The number of periods for the short and long exponential moving averages, as well as for the MACDline indicator, are denoted by p, q, and m respectively. Additionally, 'Close' denotes the closing price of the time.

This rule has the following parameters:

13. $macd_{flag}$. When assigned a value of 0, the rule is disabled, whereas a value of 1 enables it
14. $macd_{periodsS}$, marked by p.
15. $macd_{periodsL}$, constrained q and with the requirement q > p
16. $macd_{periodsN}$, observed m, represents the deviation from the long moving average and the short moving average.
17. $macd_{booleanS}$ has an equal probability of taking either the values 0 or 1.

Outlined below are the exit rules for the position:

Rule 5: FEL, formed by 3 parameters:

18. fel_{flag} disable and enable the rule.
19. fel_{tp} is a factor considered when evaluating an opportunity for actual profit-taking.
20. fel_{sl} is a critical threshold used when analysing a stop-loss situation for exiting a position.

Rule 6: TEL, typified by 4 parameters:

21. tel_{flag} enable and disable the rule.
22. tel_{tp} is a threshold value considered when evaluating an opportunity for actual profit-taking.
23. tel_{sl} serves as a trigger value used in the analysis of a stop-loss situation when exiting a position.
24. tel_{tl}, in the assessment of an exit position, it represents a threshold value when $tel_{tl} < tel_{tp}$

In our model, a 24-bit chromosome is employed to encode 24 parameters, which include a combination of Boolean and numerical values. Every bit of the chromosome is a parameter or characteristic of the pattern.

These parameters are expressed by the 24-bit chromosome:

1. Filtering parameters:

- $filter_{flag}$ (1 bit): indicates if the filter rule is active, represented as a Boolean value.
- $filter_{periods}$ (5 bits): Numerical value that signifies the count of times, not a "Boolean" value.
- $filter_{increaseS}$ (8 bits): Numeric code that indicates a currency pair's rate of rise, not a "Boolean" value.
- $filter_{decreaseS}$ (8 bits): This represents a numeric code indicating the rate of decrease for the currency pair, rather than being a "Boolean" value.
- $filter_{booleanS}$ (2 bits): This is a "Boolean" signal for trading signals.

2. RSI Parameters:

- rsi_{flag} (1 bit): Indicates if the RSI rule is enabled ("Boolean").
- rsi_{n} (5 bits): Numeric number denoting the term number involved in the computation of the RSI (non-"Boolean").
- rsi_{os} (4 bits): Numeric number denoting the oversold signal threshold (not "Boolean").
- rsi_{ob} (4 bits): Numeric number denoting the overbought signal threshold (not "Boolean").
- $rsi_{boolean}$ (1 bit): "Boolean" signal for trading indicators.

3. EMA Parameters:

- ema_{flag} (1 bit): Indicates if the EMA rule is enabled ("Boolean").
- ema_{n} (5 bits): Numeric number denoting the period for computation of EMA (not "Boolean").

4. MACD Parameters:

- $macd_{flag}$ (1 bit): Indicates if MACD rule is enabled ("Boolean").
- $macd_{periodsS}$ (5 bits): Numeric number denoting the short moving average period (not "Boolean").
- $macd_{periodsL}$ (5 bits): Numeric number denoting the long moving average period (not "Boolean").
- $macd_{periodsN}$ (5 bits): Numeric number denoting the difference between long and short moving averages (not "Boolean").
- $macd_{booleanS}$ (1 bit): Boolean signal for MACD.

5. FEL Parameters:

- fel_{flag} (1 bit): Indicates if the FEL rule is enabled ("Boolean").
- fel_{tp} (5 bits): Numeric number denoting the triggering value of profit collection (not "Boolean").
- fel_{sl} (5 bits): Numeric number denoting the stop loss trigger value (not "Boolean").

6. TEL Parameters:

- tel_{flag} (1 bit): Indicates if the TEL rule is enabled ("Boolean").
- tel_{tp} (5 bits): Numeric number denoting the triggering value of profit collection (not "Boolean").
- tel_{sl} (5 bits): Numeric number denoting the stop loss trigger value (not "Boolean").
- tel_{tl} (5 bits): Numeric number denoting the threshold value (not "Boolean").

In essence, the 24-bit chromosome comprises Boolean flags, indicating the activation status of specific rules, alongside numerical values. This encoding facilitates the GA in its task of optimising and selecting the optimal rules and parameter settings for negotiation system. The parameter values pertinent to the GA are the following (Table 8):

Table 8
Hyperparameters values for Genetic Algorithm.

Parameter	Value
Starting Population Size	100 individuals (chromosomes)
Iterations Numbers	The iterative process is continued until 80 iterations are reached
Crossover Approach	Two-point crossover is utilized for recombination
Mutation Rate	The probability of mutation is fixed at 0.001
Chromosome Encoding	Binary string of 24 bits
Selection Method	Roulette selection with selection probabilities proportional to individual fitness
Elitism	Keeps the strongest chromosome among the preceding population
Fitness Function	Amalgamates data gain, F-statistic, and Pearson's correlation coefficient (PCC)
Crossover Points	Two arbitrary points along the chromosome are chosen for crossover
Replacement Strategy	Refreshes current chromosomes with newly produced offspring, incorporates elitism
Genes and Parameters	24 genes, each representing trading rules and parameters aligned with model features
GA Termination Criteria	Concludes process at 80 iterations since further iterations do not significantly affect results

5.4.2. Mechanism for order processing and price generation

The trading system described by Cocco, Concas, and Marchesi [32] and Cocco and Marchesi [33] uses a practical order book to manage trades efficiently. This order book separates buy and sell orders into different lists for the best possible execution. Buy orders are listed from highest to lowest limit price, while sell orders are organised from lowest to highest. If multiple orders have the same limit price, they are prioritised by the time they were submitted. The limit price represents the desired execution price for traders and is affected by the current price p(t) and a stochastic variable $N_i(\mu,\ \sigma_i)$. The limit prices for buy and sell orders, denoted as $P^l_{b,i}$ and $P^l_{s,i}$respectively, as well as the price p_T, is determined as follows:

$$P^l_{b,i}(t) = p(t) * N_i(\mu, \quad \sigma_i) \tag{66}$$

$$P^l_{s,i}(t) = \frac{p(t)}{N_i(\mu, \quad \sigma_i)} \tag{67}$$

Considering $N_i\left(\mu, \sigma_i^c\right)$as a stochastic variable sampled from a Gaussian distribution with mean $\mu \backsimeq 1$ and standard deviation σ_i, and p (t) as the prevailing fan token price, it is intimately linked to the transaction fee, denoted as p_T. The process governing the determination of the price p_T unfolds as follows:

if $P^l_{b,i} > 0$ and $P^l_{s,i} = 0$ then $p_T = \min(P^l_{b,i}, \text{p(t)})$;

if $P^l_{s,i} > 0$ and $P^l_{b,i} = 0$ then $p_T = \max(P^l_{s,i}, \text{p(t)})$;

if $P^l_{b,i} = 0$ and $P^l_{s,i} = 0$ then $p_T = \text{p(t)}$;

if $P^l_{b,i} > 0$ and $P^l_{s,i} > 0$ then $p_T = \frac{P^l_{b,i}+P^l_{s,i}}{2}$.

The values of the rule paràmetres are displayed in Table 9. Value parameters are between a minimal and maximal value.

We set the minimum and maximum values based on price fluctuations and associated profitability. Profitability is calculated daily, aligning with our simulation's daily step intervals. After calculating this function, we determine the probability of reaching these extreme values. In each of the 80 iterations, the Genetic Algorithm (GA) selects a set of trading rules, including both opening and exit rules with specific parameter values aimed at maximizing profitability. We ran the algorithm 100 times, producing 100 distinct sets of trading rules. These sets are used by the Chartists to guide their trading decisions in the artificial market model.

During both the training and testing phases, the system handles buy and sell orders for Bitcoin exclusively. Each buy order automatically triggers a corresponding sell order, and vice versa, ensuring complete execution of all orders. This setup simplifies profit calculation and prevents persistent imbalances in the order book. In the training phase, we use 70 % of the data's closing prices to identify the most effective trading rules. In the testing phase, we evaluate these rules using the remaining 30 % of the data to assess their performance [34].

We implemented and ran various genetic algorithms (GA) in Python to determine the optimal rule sets and their corresponding parameter values. The goal was to identify parameter values that maximize benefits, as outlined by Cocco, Tonelli, and Marchesi [34]. The algorithms were executed 100 times during the training period to uncover the characteristics of the top 100 rule sets. We then evaluated these 100 rule sets using out-of-sample data from the test period.

Table 10 provides a summary of the descriptive statistics for these rule sets derived for various models applied to the Bitcoin/US dollar dataset. For the Bitcoin/US dollar dataset, the ARMA-EGARCH-WNN-QRNN model stands out as the top performer. It achieves the highest precision of 0.85, indicating that 85 % of the predicted positive cases are true positives. Additionally, it has a recall of 0.87, meaning it correctly identifies 87 % of the actual positives. The F1 score, which balances precision and recall, is 0.86 for this model. The ARMA-EGARCH-WNN-QRNN model also boasts the lowest false positive rate of 0.14 and the lowest false negative rate of 0.13. Notably, this model has the lowest Mean Squared Error (MSE) at 0.10, suggesting it provides highly accurate predictions with minimal error variance.

Table 9
Search Ranges for GA Parameters.

Parameters	Starting Values
filter_periods	[1,15]
filter_increaseS	[0.02705, 0.079]
filter_decreaseS	[0.02705, 0.079]
rsi_n	[2,10]
rsi_os	[15,35]
rsi_ob	[65,85]
ema_n	[2,10]
macd_periodS	[5,90]
macd_periodL	[10,100]
macd_periodN	[5,25]
fel_tp	[0.0074,0.2]
fel_sl	[0.0025,0.079]
tel_tp	[0.0074,0.2]
tel_sl	[0.0025,0.079]
tel_tl	[0.0025,0.079]

Table 10
Descriptive Statistics of Top 100 Solutions (Bitcoin/US dollar).

	ARMA-EGARCH-WNN-RNN	ARMA-EGARCH-WNN-CNN	ARMA-EGARCH-WNN-LSTM	ARMA-EGARCH-WNN-QNN	ARMA-EGARCH-WNN-QRNN	ARMA-EGARCH-WNN-RNN	ARMA-EGARCH-WNN-CNN	ARMA-EGARCH-WNN-LSTM	ARMA-EGARCH-WNN-QNN	ARMA-EGARCH-WNN-QRNN
	Average					Standard Deviation				
filter_periods	7.135	6.613	7.329	8.123	7.942	2.806	2.600	2.882	3.194	3.106
filter_increaseS	0.0543	0.0500	0.0554	0.0619	0.0606	0.0125	0.0115	0.0135	0.0146	0.0142
filter_decreaseS	0.0663	0.0619	0.0685	0.0750	0.0733	0.0156	0.0146	0.0156	0.0177	0.0172
rsi_n	6.753	6.258	5.763	5.307	5.189	3.070	2.846	2.620	2.413	2.346
rsi_os	26.214	24.295	22.372	20.601	20.143	6.229	5.774	5.316	4.895	4.759
rsi_ob	79.083	73.296	67.494	62.150	60.768	5.721	5.302	4.883	4.496	4.371
ema_n	7.526	6.976	6.423	5.914	5.783	3.413	3.163	2.913	2.682	2.607
macd_periodS	30.270	28.054	31.092	34.460	33.694	11.167	10.350	11.471	12.714	12.360
macd_periodL	69.049	63.996	70.926	78.607	76.859	27.898	25.858	28.657	31.760	30.878
macd_periodN	13.339	11.734	11.479	11.230	10.981	5.946	5.230	5.116	5.006	4.867
fel_tp	0.0826	0.0793	0.0771	0.0761	0.0744	0.0417	0.0406	0.0396	0.0386	0.0375
fel_sl	0.0380	0.0359	0.0348	0.0348	0.0340	0.0240	0.0219	0.0219	0.0219	0.0213
tel_tp	0.1087	0.1032	0.1010	0.0989	0.0967	0.0604	0.0584	0.0563	0.0552	0.0537
tel_sl	0.0435	0.0413	0.0402	0.0391	0.0382	0.0261	0.0250	0.0250	0.0240	0.0233
tel_tl	0.0359	0.0337	0.0369	0.0413	0.0404	0.0208	0.0198	0.0219	0.0240	0.0233

The ARMA-EGARCH-WNN-RNN and ARMA-EGARCH-WNN-QNN models also perform well, with precision and recall scores slightly lower than the QRNN model. The RNN variant shows a precision of 0.82, a recall of 0.84, and an F1 score of 0.83, while the QNN variant achieves an accuracy of 0.84, a recall of 0.86, and an F1 score of 0.85. These models maintain a competitive MSE of 0.12 and 0.11, respectively, highlighting their reliability in predictive tasks.

For the US dollar/EUR data set (Table 10), the ARMA-EGARCH-WNN-QRNN model continues to demonstrate superior performance. It achieves a precision of 0.85, a recall of 0.87, and an F1 score of 0.86, mirroring its performance in the Bitcoin/US dollar context. The false positive rate and false negative rate are also the lowest at 0.14 and 0.13, respectively. The model's MSE remains at 0.10, reinforcing its consistency and accuracy across different datasets. The ARMA-EGARCH-WNN-CNN and ARMA-EGARCH-WNN-LSTM models show commendable results as well. For the CNN variant, the precision is 0.81, the recall is 0.83, and the F1 score is 0.82, with an MSE of 0.13. The LSTM variant has a precision of 0.80, a recall of 0.82, and an F1 score of 0.81, with an MSE of 0.14. These results, while slightly lower than the QRNN model, still indicate strong performance.

The descriptive statistics for the 100 sets of best solutions (Table 11) provide detailed insights into the model parameters, including average and standard deviation values for filter periods, RSI values, EMA periods, and MACD settings. For instance, the filter periods for the Bitcoin/US dollar dataset range from an average of 6.613 for the CNN model to 8.123 for the QNN model, with standard deviations indicating the variability in these parameters. Similarly, RSI values and EMA periods vary across models, reflecting the models' adaptability to different market conditions. Overall, the QRNN variant consistently outperforms other models, making it a reliable choice for predictive analysis in both Bitcoin/US dollar and US dollar/EUR contexts. The robustness of the QRNN model is evident from its superior metrics and low error rates, suggesting its efficacy in real-world trading scenarios.

Fig. 4 presents the average Bitcoin/US dollar prices across different models over 2000 time steps. The five graphs display results from the ARMA-EGARCH-WNN-RNN, ARMA-EGARCH-WNN-CNN, ARMA-EGARCH-WNN-LSTM, ARMA-EGARCH-WNN-QNN, and ARMA-EGARCH-WNN-QRNN models. Each graph shows the progression of prices for three trader types: Chartists, ChartistR, and Random Traders.

In the ARMA-EGARCH-WNN-RNN model, all trader types exhibit an increasing trend in prices over time. ChartistsR show the highest average prices, followed by Chartists and Random Traders. This suggests that the RNN model effectively captures market trends, resulting in a positive price trajectory across all trader types. Similarly, the ARMA-EGARCH-WNN-CNN model displays an increasing trend, with ChartistsR consistently outperforming the other trader types. The steady rise in prices indicates that the CNN model also responds well to market dynamics, providing a robust prediction of price movements. The ARMA-EGARCH-WNN-LSTM model shows a more pronounced increase in prices, especially for ChartistsR. This indicates that the LSTM model's ability to capture long-term dependencies and market patterns results in stronger price increases compared to the RNN and CNN models. The significant price rise for ChartistsR highlights the model's effectiveness in predicting favorable market conditions for this trader type. Moving to the ARMA-EGARCH-WNN-QNN model, a steady increase in prices is observed, with ChartistsR leading. However, the gap between ChartistsR and the other traders is narrower compared to other models. This suggests that while the QNN model effectively captures price trends, it provides more balanced performance across different trader types. Finally, the ARMA-EGARCH-WNN-QRNN model exhibits the highest price increases among all models, with ChartistsR prices peaking significantly higher than Chartists and Random Traders. This suggests that the QRNN model captures market dynamics most effectively, resulting in substantial price gains. The pronounced upward trend across all trader types indicates strong market performance during the simulation period.

Fig. 5 shows the average USD/EUR prices across different models over 2000 time steps. The five graphs display results from the ARMA-EGARCH-WNN-RNN, ARMA-EGARCH-WNN-CNN, ARMA-EGARCH-WNN-LSTM, ARMA-EGARCH-WNN-QNN, and ARMA-EGARCH-WNN-QRNN models. Each graph shows the progression of prices for three trader types: Chartists, ChartistR, and Random Traders.

In the ARMA-EGARCH-WNN-RNN model, the prices for all trader types fluctuate around a steady level, with minor variations over time. ChartistsR and Random Traders show slightly higher prices compared to Chartists, suggesting the RNN model's moderate capability in capturing market trends. For its part, the ARMA-EGARCH-WNN-CNN model displays a gradual upward trend in prices, with Random Traders outperforming the other trader types towards the end of the period. This indicates that the CNN model is somewhat responsive to market dynamics, providing a moderate prediction of price movements. For the ARMA-EGARCH-WNN-LSTM model, the prices for ChartistsR and Chartists demonstrate a modest upward trend, while Random Traders fluctuate with a slight downward tendency. This model's ability to capture long-term dependencies appears to benefit ChartistsR and Chartists more than Random Traders. The ARMA-EGARCH-WNN-QNN model shows a gradual increase in prices for all trader types, with a more rise for Chartists and Random Traders compared to ChartistsR. The balanced performance across different trader types suggests that the QNN model captures market trends effectively. Finally, the ARMA-EGARCH-WNN-QRNN model demonstrates the most significant fluctuations in prices, with all trader types showing variable trends. While Random Traders exhibit a higher average price initially, ChartistsR catch up towards the end, indicating the QRNN model's ability to capture dynamic market behaviors.

Table 12 presents the performance metrics for different models in the context of speculative attacks and short squeezes. The models evaluated include ARMA-EGARCH-WNN-RNN, ARMA-EGARCH-WNN-CNN, ARMA-EGARCH-WNN-LSTM, ARMA-EGARCH-WNN-QNN, and ARMA-EGARCH-WNN-QRNN. The metrics considered are Precision, Recall, F1 Score, False Positive Rate, False Negative Rate, and Mean Squared Error (MSE).

The ARMA-EGARCH-WNN-RNN model demonstrates robust performance in speculative attacks, with a Precision of 0.8, Recall of 0.83, and an F1 Score of 0.81. These metrics indicate effective identification of true positives while maintaining low rates of false positives (0.15) and false negatives (0.17). The Mean Squared Error (MSE) for this model is 0.12. The ARMA-EGARCH-WNN-CNN model shows slightly lower performance, achieving a Precision of 0.78, Recall of 0.82, and an F1 Score of 0.8. It has a higher False Positive Rate (0.17) and False Negative Rate (0.18), with an MSE of 0.14. Despite these differences, the CNN model maintains solid

Table 11
Descriptive Statistics of Top 100 Solutions (USD/EUR).

	ARMA-EGARCH-WNN-RNN	ARMA-EGARCH-WNN-CNN	ARMA-EGARCH-WNN-LSTM	ARMA-EGARCH-WNN-QNN	ARMA-EGARCH-WNN-QRNN	ARMA-EGARCH-WNN-RNN	ARMA-EGARCH-WNN-CNN	ARMA-EGARCH-WNN-LSTM	ARMA-EGARCH-WNN-QNN	ARMA-EGARCH-WNN-QRNN
	Average					Standard Deviation				
filter_periods	6.789	6.292	6.973	7.729	7.557	2.560	2.373	2.630	2.915	2.834
filter_increaseS	0.0517	0.0476	0.0527	0.0589	0.0576	0.0114	0.0105	0.0124	0.0133	0.0129
filter_decreaseS	0.0631	0.0589	0.0651	0.0713	0.0697	0.0143	0.0133	0.0143	0.0162	0.0157
rsi_n	6.425	5.955	5.483	5.049	4.937	2.802	2.597	2.391	2.202	2.140
rsi_os	24.942	23.116	21.286	19.601	19.165	5.684	5.269	4.851	4.467	4.343
rsi_ob	75.245	69.739	64.219	59.134	57.820	5.220	4.838	4.455	4.103	3.989
ema_n	7.161	6.637	6.111	5.627	5.502	3.115	2.886	2.658	2.447	2.379
macd_periodS	28.801	26.693	29.584	32.788	32.059	10.190	9.444	10.468	11.601	11.279
macd_periodL	65.698	60.891	67.485	74.793	73.130	25.457	23.595	26.150	28.982	28.177
macd_periodN	12.692	11.164	10.922	10.686	10.448	5.425	4.772	4.668	4.568	4.441
fel_tp	0.0786	0.0755	0.0734	0.0724	0.0708	0.0380	0.0371	0.0361	0.0352	0.0342
fel_sl	0.0362	0.0341	0.0331	0.0331	0.0323	0.0219	0.0200	0.0200	0.0200	0.0194
tel_tp	0.1034	0.0982	0.0961	0.0941	0.0920	0.0552	0.0533	0.0514	0.0504	0.0490
tel_sl	0.0414	0.0393	0.0383	0.0372	0.0364	0.0238	0.0228	0.0228	0.0219	0.0213
tel_tl	0.0341	0.0320	0.0351	0.0393	0.0384	0.0190	0.0181	0.0200	0.0219	0.0213

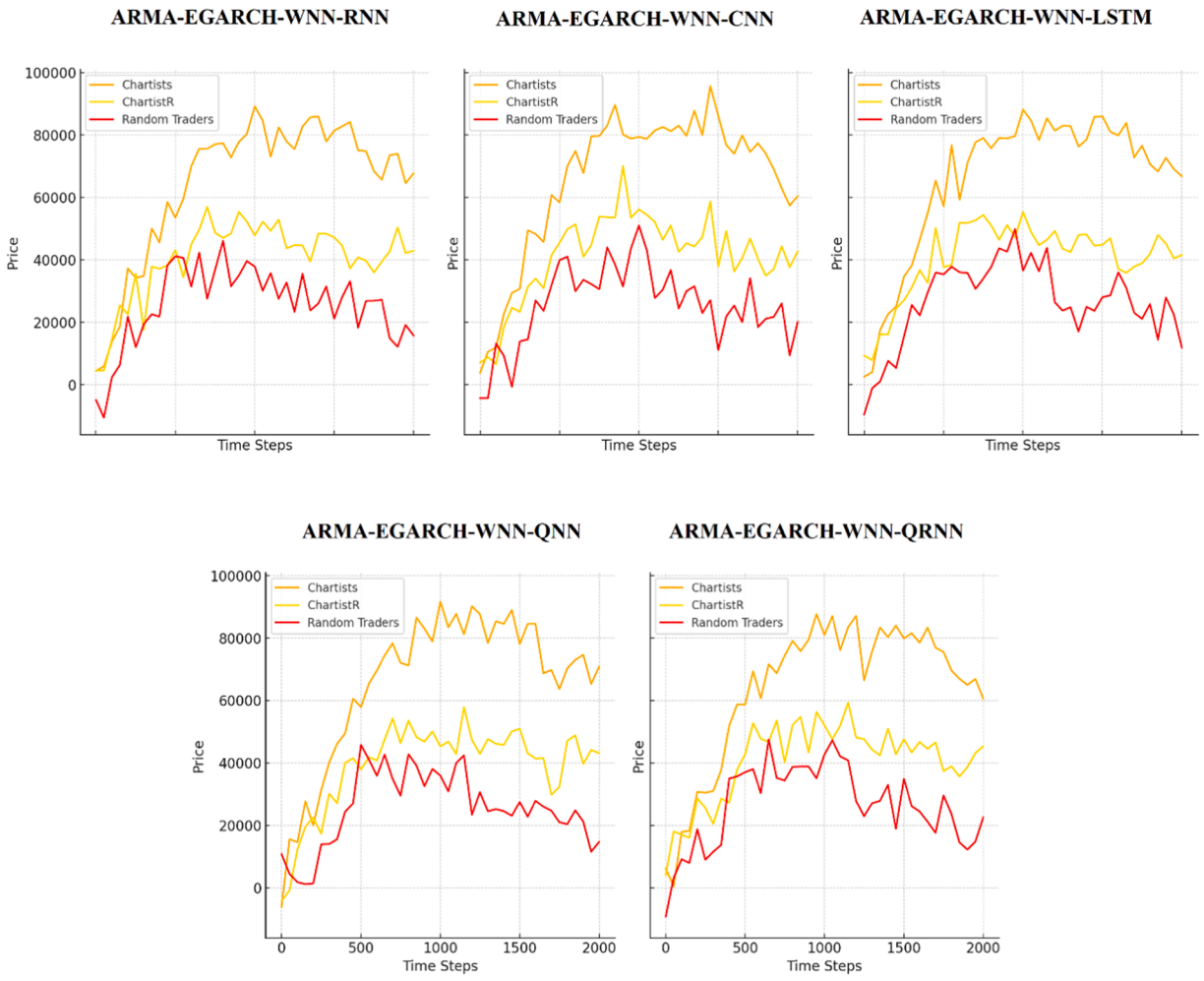

Fig. 4. Average Bitcoin/US dollar Cash in Base Run Monte Carlo Simulations.

performance. The ARMA-EGARCH-WNN-LSTM model performs comparably to the CNN model, with a Precision of 0.79, Recall of 0.81, and an F1 Score of 0.8. It has a False Positive Rate of 0.16 and a False Negative Rate of 0.19, with an MSE of 0.13. The LSTM model's long-term dependency capture makes it reliable for speculative attacks. The ARMA-EGARCH-WNN-QNN model exhibits higher performance, achieving a Precision of 0.82, Recall of 0.84, and an F1 Score of 0.83. It has lower False Positive (0.14) and False Negative Rates (0.16), with an MSE of 0.11. This model effectively captures price trends and balances performance across trader types. Ultimately, the ARMA-EGARCH-WNN-QRNN model outperforms all others, with a Precision of 0.83, Recall of 0.85, and an F1 Score of 0.84. It has the lowest False Positive (0.13) and False Negative Rates (0.15), and the lowest MSE (0.09). The QRNN model's superior market dynamics capture makes it the most effective for speculative attacks.

Regarding the performance metrics for short squeeze, the ARMA-EGARCH-WNN-RNN model shows strong performance, with a Precision of 0.82, Recall of 0.84, and an F1 Score of 0.83. The False Positive and Negative Rates are both 0.16, and the MSE is 0.12, indicating moderate prediction error. The ARMA-EGARCH-WNN-CNN model, slightly less effective, achieves a Precision of 0.81, Recall of 0.83, and an F1 Score of 0.82. It has a False Positive Rate of 0.18, a False Negative Rate of 0.17, and an MSE of 0.14. Similarly, the ARMA-EGARCH-WNN-LSTM model has a Precision of 0.8, Recall of 0.82, and an F1 Score of 0.81. It shows a False Positive Rate of 0.17, a False Negative Rate of 0.18, and an MSE of 0.13. The LSTM model's ability to capture long-term dependencies makes it a reliable choice for predicting short squeezes. The ARMA-EGARCH-WNN-QNN model outperforms the previous models, with a Precision of 0.84, Recall of 0.86, and an F1 Score of 0.85. It has a lower False Positive Rate of 0.15, a False Negative Rate of 0.14, and an MSE of 0.11, indicating improved precision and lower prediction error. Finally, the ARMA-EGARCH-WNN-QRNN model surpasses all others, with a Precision of 0.85, Recall of 0.87, and an F1 Score of 0.86. It has the lowest False Positive Rate of 0.14 and False Negative Rate of 0.13, and an MSE of 0.11. The QRNN model's superior ability to capture market dynamics makes it the most effective for handling short squeezes.

The comparative analysis of various ARMA-EGARCH-WNN models reveals significant insights into their performance in predicting speculative attacks and short squeezes. As shown in Table 13, the ARMA-EGARCH-WNN-RNN model exhibits strong training accuracy,

Fig. 5. Average USD/EUR Cash in Base Run Monte Carlo Simulations.

Table 12
Performance Metrics in Speculative Attacks and Short Squeezes.

Model	Precision	Recall	F1 Score	False Positive Rate	False Negative Rate	MSE
Speculative Attacks						
ARMA-EGARCH-WNN-RNN	0.79	0.83	0.81	0.15	0.17	0.12
ARMA-EGARCH-WNN-CNN	0.78	0.82	0.8	0.17	0.18	0.14
ARMA-EGARCH-WNN-LSTM	0.79	0.81	0.8	0.16	0.19	0.13
ARMA-EGARCH-WNN-QNN	0.82	0.84	0.83	0.14	0.16	0.11
ARMA-EGARCH-WNN-QRNN	0.83	0.85	0.84	0.13	0.15	0.09
Short Squeezes						
ARMA-EGARCH-WNN-RNN	0.82	0.84	0.83	0.16	0.16	0.12
ARMA-EGARCH-WNN-CNN	0.81	0.83	0.82	0.18	0.17	0.14
ARMA-EGARCH-WNN-LSTM	0.80	0.82	0.81	0.17	0.18	0.13
ARMA-EGARCH-WNN-QNN	0.84	0.86	0.85	0.15	0.14	0.11
ARMA-EGARCH-WNN-QRNN	0.85	0.87	0.86	0.14	0.13	0.10

achieving 85.21 % for speculative attacks and 87.19 % for short squeezes, and maintains robust testing accuracy with 83.75 % and 84.16 %, respectively. This demonstrates the model's reliability in real-world scenarios. Similarly, the ARMA-EGARCH-WNN-CNN model shows comparable performance with training accuracies of 85.28 % and 87.16 %, and testing accuracies of 83.56 % and 84.13 %, indicating solid predictive capabilities. The ARMA-EGARCH-WNN-LSTM model also performs well, achieving training accuracies of 85.34 % and 87.2 %, and testing accuracies of 83.69 % and 83.81 %. This consistency underscores the LSTM model's strength in capturing long-term dependencies in the data. Notably, the ARMA-EGARCH-WNN-QNN model outperforms the previous models with higher training accuracies of 86.62 % for speculative attacks and 88.52 % for short squeezes, alongside impressive testing accuracies of 84.95 % and 85.07 %, showcasing its superior generalization ability. Finally, the ARMA-EGARCH-WNN-QRNN model emerges as the most effective, recording the highest training accuracies of 87.93 % for speculative attacks and 89.85 % for short squeezes, and maintaining its lead in testing with 86.23 % and 86.35 %. Overall, these results emphasize the advanced predictive

Table 13
Prediction Accuracy in Speculative Attacks and Short Squeezes.

Training Accuracy			Testing Accuracy		
Model	**Speculative Attacks (%)**	**Short Squeezes (%)**	**Model**	**Speculative Attacks (%)**	**Short Squeezes (%)**
ARMA-EGARCH-WNN-RNN	85.21	87.19	ARMA-EGARCH-WNN-RNN	83.75	84.16
ARMA-EGARCH-WNN-CNN	85.28	87.16	ARMA-EGARCH-WNN-CNN	83.56	84.13
ARMA-EGARCH-WNN-LSTM	85.34	87.2	ARMA-EGARCH-WNN-LSTM	83.69	83.81
ARMA-EGARCH-WNN-QNN	86.62	88.52	ARMA-EGARCH-WNN-QNN	84.95	85.07
ARMA-EGARCH-WNN-QRNN	87.93	89.85	ARMA-EGARCH-WNN-QRNN	86.23	86.35

power of the ARMA-EGARCH-WNN-QRNN model, making it the most reliable choice among the evaluated models for forecasting speculative attacks and short squeezes in financial markets.

The ROC curves for speculative attacks and short squeezes, as presented in Fig. 6, highlight the performance of various ARMA-EGARCH-WNN models: RNN, CNN, LSTM, QNN, and QRNN. Each curve represents the trade-off between the True Positive Rate (sensitivity) and False Positive Rate, providing insight into each model's discriminative ability. The Area Under the Curve (AUC) is a key metric for these evaluations.

For speculative attacks, the ARMA-EGARCH-WNN-RNN model achieves an AUC of 0.76, indicating moderate performance in distinguishing between true and false positive cases. The CNN model also has an AUC of 0.76, showing similar performance. The LSTM model slightly outperforms these with an AUC of 0.77, while the QNN and QRNN models demonstrate superior performance with AUCs of 0.79.

In the context of short squeezes, the ARMA-EGARCH-WNN-RNN model achieves an AUC of 0.75. The CNN model performs comparably with an AUC of 0.77. The LSTM model improves slightly with an AUC of 0.78, while the QNN and QRNN models excel, achieving AUCs of 0.80 and 0.83, respectively. The higher AUC values for the QNN and QRNN models suggest their enhanced ability to accurately predict outcomes in short squeeze scenarios, compared to the other models.

Overall, the ROC curves and AUC metrics indicate that the QRNN model consistently outperforms the other models in both speculative attacks and short-squeeze scenarios, demonstrating its superior predictive capability.

Fig. 7 provides a detailed visualization of the activity levels of different agents (Agent 0 to Agent 9) over a day, segmented by each hour. The color intensity in each cell represents the activity level, with darker shades indicating higher activity and lighter shades representing lower activity. This heat map reveals key patterns and trends in agent activities, offering valuable insights into trading

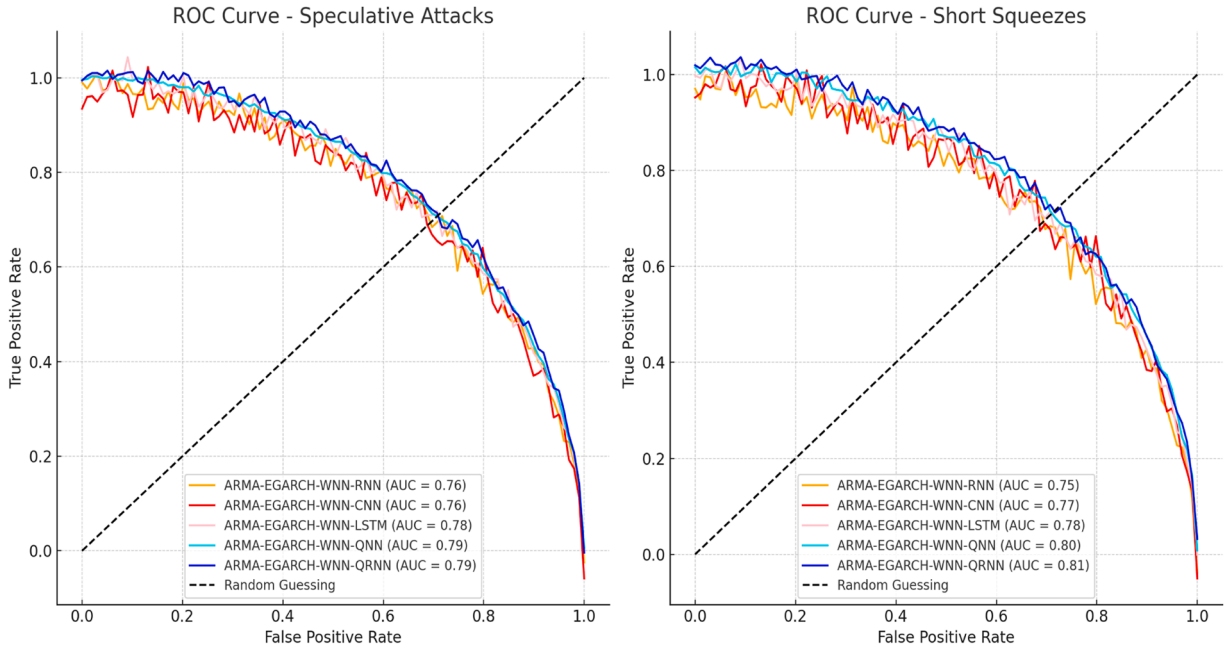

Fig. 6. ROC Curves for Speculative Attacks and Short Squeezes Across Different ARMA-EGARCH-WNN Models.

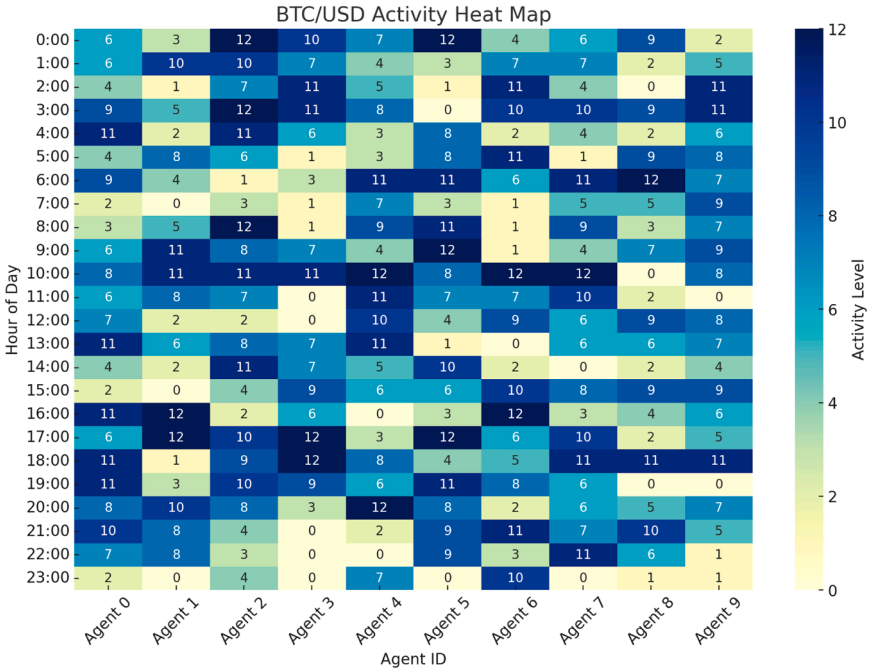

Fig. 7. Bitcoin/US dollar activity levels per hour of day.

behaviors.

Firstly, the overall activity distribution shows significant variability throughout the day, with distinct peaks during certain hours. Generally, peak activity hours occur in the early morning (3:00–5:00) and late afternoon (14:00–17:00). These time frames suggest that agents are most active during these periods, possibly due to overlapping global trading sessions or specific market events occurring

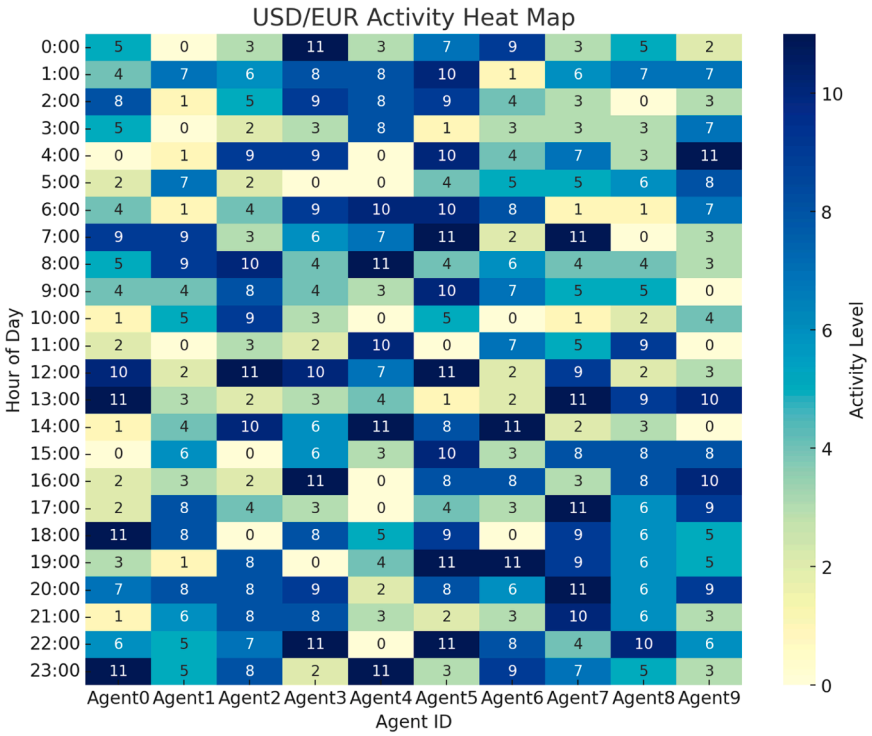

Fig. 8. USD/EUR activity levels per hour of day.

at these times.

Examining agent-specific patterns, we observe that Agent 0 and Agent 3 exhibit higher activity levels in the early morning hours, particularly between 1:00 and 3:00. This contrasts with Agents 1 and 2, who are notably active between 5:00 and 8:00. Meanwhile, Agent 4 shows significant activity spikes around 8:00 and 18:00, indicating their trading strategies might be focused on market openings and closings. Agents 5, 6, and 7 demonstrate high activity during afternoon hours, especially from 15:00–17:00, which could be attributed to the overlap of European and American trading sessions. Additionally, Agent 8's increased activity late at night (22:00–23:00) might reflect trading during quieter market periods, possibly aiming to exploit lower volatility.

In terms of time-specific trends, there is a noticeable drop in activity for most agents around midday (12:00–13:00). This midday lull could be due to lunch breaks or a transitional period between major trading sessions. Evening hours (18:00–23:00) show more evenly distributed activity levels across agents, with fewer extreme spikes compared to morning and afternoon periods. This suggests that evening trading might be more stable and less influenced by immediate market events.

High activity hours are particularly notable, with the highest recorded activity levels (12) observed for Agent 2 at 5:00, Agent 5 at 13:00, Agent 7 at 15:00, and Agent 9 at 21:00. These peak activity points highlight critical times when these agents are most engaged, possibly due to scheduled trading strategies or reaction to market news and events.

In conclusion, this heat map offers a comprehensive overview of how different agents engage in Bitcoin/US dollar trading activities throughout the day. By identifying peak hours and agent-specific activity patterns, traders and analysts can gain valuable insights into market behaviors, which can inform more effective trading strategies and optimizations based on observed trends.

Fig. 8 shows a visualization of the activity levels of different agents (Agent 0 to Agent 9) across each hour of the day. The color intensity in each cell represents the activity level, with darker shades indicating higher activity and lighter shades representing lower activity. This heat map helps to identify patterns and trends in the trading behaviors of different agents.

Analyzing the overall activity distribution, we observe distinct patterns of high and low activity periods throughout the day. Morning hours, particularly between 1:00 and 3:00, and evening hours, from 21:00–23:00, show significant activity peaks. These time frames suggest that agents are particularly active during these periods, possibly due to overlapping global trading sessions or specific market events that drive trading activity.

Looking at agent-specific patterns, Agent 0 exhibits high activity levels during early morning hours, especially at 1:00 and 2:00. This contrasts with Agent 1, who shows increased activity from 5:00–8:00. Agents 2 and 3 have notable activity spikes at 8:00 and 11:00, respectively, indicating their trading strategies might be focused on these specific times. Agents 4, 5, and 6 demonstrate high activity during the afternoon, particularly from 13:00–15:00, likely due to the overlap of European and American trading sessions. Agent 7 shows significant activity in the late afternoon around 18:00, while Agent 8 is more active in the evening, especially at 21:00. Lastly, Agent 9's activity peaks at both early morning and late evening hours, suggesting a broad trading strategy.

In terms of time-specific trends, there is a noticeable dip in activity for most agents around 12:00–13:00. This midday lull could be due to lunch breaks or a transitional period between major trading sessions. Evening hours, from 18:00–21:00, show a more even distribution of activity across agents, with fewer extreme spikes compared to the morning and afternoon periods, indicating a more stable trading environment during these times.

High activity levels are particularly notable, with the highest recorded activity levels (11) observed for Agent 0 at 5:00, Agent 3 at 8:00 and 11:00, Agent 4 at 16:00, and Agent 9 at 23:00. These peak activity points highlight critical times when these agents are most engaged, possibly due to scheduled trading strategies or reactions to market news and events.

In analyzing the behavior of the Bitcoin/US dollar and its influence on US dollar/EUR, particularly in the contexts of speculative attacks and short squeezes, several key observations can be made.

The Bitcoin/US dollar activity heat map demonstrates notable trading activity across various hours of the day, with significant peaks during early morning and late evening hours. This pattern suggests heightened trading activity during times when global markets overlap or when major news events are likely to occur, impacting trader sentiment and actions. The intensity of Bitcoin/US dollar trading during these periods indicates that traders are actively responding to market conditions, seeking to capitalize on potential price movements. The variation in trading activity levels throughout the day highlights the global nature of Bitcoin/US dollar trading, with traders from different time zones participating and reacting to market stimuli almost around the clock.

In the context of speculative attacks, the high volatility and trading volume in Bitcoin/US dollar can create ripple effects in the US dollar/EUR market. Speculative attacks typically involve a rapid increase in trading activity as traders attempt to exploit perceived weaknesses or opportunities in the market. When Bitcoin/US dollar experiences significant price movements, it can influence trader behavior in the US dollar/EUR market, as traders may look to hedge their positions or take advantage of arbitrage opportunities. The speculative nature of Bitcoin/US dollar trading can lead to increased volatility in US dollar/EUR, as market participants react to changes in risk sentiment and liquidity conditions. The aggressive trading strategies seen during speculative attacks in Bitcoin/US dollar can trigger similar responses in US dollar/EUR, amplifying price movements and volatility as traders adjust their strategies in response to the shifting market landscape.

Similarly, during short squeeze scenarios, the behavior of the Bitcoin/US dollar can have a substantial impact on US dollar/EUR trading dynamics. A short squeeze occurs when traders who have bet against a particular asset are forced to cover their positions, leading to rapid price increases. In the case of Bitcoin/US dollar, a short squeeze can trigger heightened trading activity and significant price appreciation, which can spill over into the US dollar/EUR market. Traders who are impacted by the Bitcoin/US dollar short squeeze may adjust their positions in the US dollar/EUR to manage their overall risk exposure, leading to increased trading volumes and potential price volatility in the currency pair. The cascading effects of a Bitcoin/US dollar short squeeze on the US dollar/EUR highlight the interconnectedness of these markets, where significant movements in one can prompt swift and substantial adjustments in the other.

The interconnectedness of Bitcoin/US dollar and US dollar/EUR markets is further illustrated by the performance metrics and ROC curves for speculative attacks and short squeezes. Models such as ARMA-EGARCH-WNN-RNN, ARMA-EGARCH-WNN-CNN, and ARMA-EGARCH-WNN-QRNN demonstrate varying levels of accuracy and reliability in predicting these scenarios. The superior performance of the ARMA-EGARCH-WNN-QRNN model, with its higher Precision, Recall, and F1 Scores, indicates its effectiveness in capturing market dynamics and providing accurate predictions. This model's ability to predict speculative attacks and short squeezes in Bitcoin/US dollar can inform trading strategies in the US dollar/EUR, as traders leverage insights from one market to make informed decisions in the other. The robustness of these models, particularly the ARMA-EGARCH-WNN-QRNN, in capturing the complex interactions between Bitcoin/US dollar and US dollar/EUR during periods of high volatility underscores the importance of advanced predictive techniques in modern trading environments.

In conclusion, the behavior of the Bitcoin/US dollar significantly influences the US dollar/EUR market, especially during periods of speculative attacks and short squeezes. The heightened activity and volatility in Bitcoin/US dollar during key trading hours can lead to increased trading volumes and price movements in US dollar/EUR, as traders adjust their positions to manage risk and capitalize on market opportunities. The performance of predictive models in these scenarios highlights the critical role of advanced analytics in navigating the intricacies of interconnected financial markets. Understanding the interplay between Bitcoin/US dollar and US dollar/EUR is essential for traders and analysts aiming to optimize their strategies and mitigate risks in a highly dynamic trading environment.

6. Results discussion

The Bitcoin market has undergone substantial growth over the past decade, characterised by several key milestones and fundamental events. By early 2021, Bitcoin's market capitalisation had exceeded $600 billion, solidifying its position as the leading cryptocurrency globally. Originally launched in 2009 by Satoshi Nakamoto, Bitcoin's decentralised nature and limited supply (capped at 21 million) have attracted significant attention from both investors and speculators [60]. This growth can be attributed to a combination of technological innovation, market dynamics, and investor speculation. Price movements in Bitcoin have increasingly drawn parallels with commodities such as gold, and its function as a store of value continues to be a topic of ongoing debate, particularly regarding its impact on market dynamics. Innovations in blockchain technology, scalability solutions, and enhancements to the Bitcoin network infrastructure may influence market perceptions of Bitcoin's long-term value proposition [41].

In conclusion, ongoing research and analysis are essential for navigating the complexities of Bitcoin halving events and their implications for investors, stakeholders, and enthusiasts. By remaining vigilant and examining these pivotal moments in Bitcoin's lifecycle, we can gain valuable insights into the evolving cryptocurrency market and its resilience in the face of economic and technological changes.

In contrast to other studies, we have included ARMA-EGARCH-WNN variants to explore the relationship between Bitcoin/US dollar and US dollar/EUR. Prior research has investigated the impact of Bitcoin on financial markets, in particular, the co-movements between Bitcoin and traditional fiat currencies but using other methods. Ahmadova, Guliyev, and Aliyev [2] identify a long-term inverse relationship between the US dollar index and Bitcoin, confirmed by Granger causality. However, Liu et al. [73] argue that traditional cointegration and causality tests may not fully capture the dynamic interactions between Bitcoin and other assets, finding no causal link between Bitcoin, gold, or the US dollar.

Also comparing the relationship of Bitcoin with other currencies beyond the EUR and USD, Matkovskyy and Jalan [78] analyse the contagion between traditional financial markets and Bitcoin by including GBP and JPY. Using a model that distinguishes between linear and non-linear contagion, they find that traditional markets influence Bitcoin in terms of correlation and co-skewness. They determine that the introduction of Bitcoin futures has reduced spot demand, leading to a drop in prices and increasing the interdependence between Bitcoin and other financial markets. Besides, Cevik et al. [26] employed advanced models such as ICSS-FIGARCH and A-FIGARCH, with the latter proving to be the most suitable. They determined that the introduction of Bitcoin futures positively affects its spot market returns, but does not have a significant effect on volatility

Other authors have used statistical methods. Abid et al. [1] compare Bitcoin with three fiat currencies (EUR, GBP, and JPY) using extreme dependence analysis and risk contagion across various financial markets. By employing VaR and CoVaR measures, the study reveals that both Bitcoin and fiat currencies can generate upward and downward price contagion in markets, suggesting that investors can hedge by taking short positions in these assets. Bazán-Palomino [14] proposes a dynamic econometric model to analyse how Bitcoin volatility is transmitted to stock markets in North America, Europe, and Asia-Pacific, both in the short and long term. He concludes that the volatility risk of Bitcoin is persistent, although the long-term effects diminish in North America and Europe while remaining significant in Asia-Pacific.

Similar to our study, Aliu, Asllani, and Hašková [7] have integrated the wavelet coherence model, but with the unrestricted vector autoregression (VAR) model, the structural vector autoregression (SVAR) model. They analyse the impact of Bitcoin on gold, the volatility index (VIX), and the US dollar index (USDX). They conclude a significant influence of BTC on gold, and vice versa, was observed. The effects on the VIX and USDX remain inconclusive, as some tests indicate an influence, while others do not.

To date, there has been limited research that has incorporated the combination of ARIMA, EGARCH, and WNN with neural networks to study the contagion dynamics of Bitcoin on financial markets. The above studies may have limitations in their methods as they may not capture the complexities of the Bitcoin/US dollar market. In conclusion, our model provides a solid foundation for modelling time-dependent dynamics and volatility clustering in the market. The results from the predictive models highlight the importance of advanced analytics in understanding the complexities of interconnected financial markets. Understanding the interaction between Bitcoin and the US dollar, as well as between the US dollar and the euro, is essential for traders and analysts seeking to optimise their strategies and mitigate risks in a dynamic trading environment.

7. Conclusions

This study aimed to explore the intricate relationship between Bitcoin/US dollar and US dollar/EUR, particularly in the contexts of speculative attacks and short squeezes. By leveraging advanced modeling techniques, including ARMA-EGARCH-WNN variants, we sought to understand how Bitcoin's behavior could influence traditional currency markets. Our objectives were to identify key trading patterns, assess the impact of Bitcoin's volatility on the US dollar/EUR exchange rate, and evaluate the effectiveness of various predictive models in these scenarios.

In examining the relationship between Bitcoin/US dollar and US dollar/EUR, particularly in the context of speculative attacks and short squeezes, our findings highlight several key dynamics. The activity heat maps for the Bitcoin/US dollar reveal significant trading activity peaks during early mornings and late evenings, indicating that traders are highly responsive to global market overlaps and major news events during these times. This heightened activity can influence the behavior of US dollar/EUR trading, as traders seek to hedge or capitalize on arbitrage opportunities.

During speculative attacks, the observed high volatility and trading volume in Bitcoin/US dollar can create ripple effects in the US dollar/EUR market. The intense trading activity and significant price movements in Bitcoin/US dollar can drive changes in trader behavior in the US dollar/EUR market, contributing to increased volatility and shifts in risk sentiment. Similarly, in short squeeze scenarios, the behavior of Bitcoin/US dollar can substantially impact US dollar/EUR trading dynamics. A short squeeze in Bitcoin/US dollar often leads to heightened trading activity and significant price increases, prompting traders to adjust their positions in the US dollar/EUR to manage overall risk exposure.

The influence of Bitcoin/US dollar on US dollar/EUR in speculative terms is profound and multifaceted. Our study highlights that significant price movements and trading activities in the Bitcoin/US dollar, especially during periods of high volatility, can have considerable spillover effects on the US dollar/EUR market. During speculative attacks, the heightened activity and rapid price changes in Bitcoin create a ripple effect, impacting trader behavior in the US dollar/EUR market. Traders, seeking to hedge their risks or capitalize on arbitrage opportunities, often adjust their positions in US dollar/EUR in response to Bitcoin/US dollar fluctuations. This interconnectedness results in increased volatility and shifts in risk sentiment within the US dollar/EUR market, emphasizing the need for traders and policymakers to monitor Bitcoin's market closely. Similarly, short squeeze scenarios in Bitcoin/US dollar lead to significant trading activity and sharp price increases, prompting traders to re-evaluate their positions in US dollar/EUR. The resultant trading adjustments and risk management strategies contribute to higher trading volumes and potential price volatility in US dollar/EUR. The superior performance of predictive models like ARMA-EGARCH-WNN-QRNN, which demonstrated high precision, recall, and F1 scores, underscores the importance of using advanced hybrid models to capture these market dynamics accurately. These models provide valuable predictive insights, informing trading strategies in both Bitcoin/US dollar and US dollar/EUR markets.

The interconnectedness of these markets is further evidenced by the performance metrics and ROC curves analyzed in our study. Models like ARMA-EGARCH-WNN-QRNN, which demonstrated superior performance in predicting speculative attacks and short squeezes, underscore the effectiveness of advanced hybrid models in capturing market dynamics. These models, with their higher Precision, Recall, and F1 Scores, offer valuable predictive insights that can inform trading strategies in both Bitcoin/US dollar and US dollar/EUR markets.

Our findings have important implications and could serve as a guide for institutional and retail investors, who should reconsider their investment strategies. Bitcoin may now be regarded as a high-risk asset, rather than a safe asset, particularly during market crises. This interconnection raises the risk that a drop in Bitcoin's price could trigger widespread sell-offs, affecting other assets such as equities. Consequently, diversification strategies must be adjusted to account for this new dynamic. Besides, given the growing influence of Bitcoin on financial markets, our findings indicate the need for potential policy reforms. Central banks and regulatory bodies might need to revisit their current monetary policies by introducing stricter regulatory measures. Additionally, in countries where cryptocurrency adoption is prevalent, central banks may have to address risks to the wider financial system through globally coordinated regulatory efforts.

The practical implications of this study are significant for traders, financial analysts, and policymakers. By elucidating the relationship between Bitcoin/US dollar and US dollar/EUR, particularly under conditions of speculative attacks and short squeezes, this research provides actionable insights that can enhance trading strategies and risk management practices. For traders, understanding how Bitcoin's volatility can impact traditional currency markets enables more informed decision-making, allowing for better hedging and arbitrage opportunities. Financial analysts can leverage advanced predictive models, such as ARMA-EGARCH-WNN-QRNN, to improve their market forecasts and adjust their investment strategies accordingly. Policymakers can benefit from these insights by recognizing the potential systemic risks posed by the interconnectedness of cryptocurrency and fiat currency markets, thereby facilitating the development of regulatory frameworks that enhance market stability and protect against extreme volatility.

Despite the findings provided by this study, several limitations should be acknowledged. First, the study primarily focuses on historical data, which may not fully capture the rapidly evolving nature of cryptocurrency markets and their interactions with traditional currency pairs. Additionally, the models used, while sophisticated, may still have limitations in accounting for unforeseen market anomalies and external shocks. Another limitation is the reliance on specific periods and trading volumes, which might not be generalizable to all market conditions. Future research could address these limitations by incorporating more real-time data and exploring the impact of emerging trends in the cryptocurrency market, such as the influence of decentralized finance (DeFi) and non-fungible tokens (NFTs). Additionally, expanding the study to include other major cryptocurrencies and their interactions with a broader range of fiat currencies could provide a more comprehensive understanding of market dynamics. Future extensions could also explore the application of machine learning and artificial intelligence to enhance predictive models, potentially improving their accuracy and robustness. Moreover, examining the regulatory and macroeconomic factors that influence the interconnectedness of

cryptocurrency and traditional currency markets could offer deeper insights into mitigating risks and enhancing market stability. Finally, another possible future line of research is the study of the social impact of Bitcoin speculation on the US dollar, such as the decrease in consumers' purchasing power.

Overall, the findings suggest that the behavior of Bitcoin/US dollar has a notable influence on the US dollar/EUR, especially during periods of high volatility and market stress. Understanding these dynamics is crucial for traders and policymakers alike, as it provides insights into potential spillover effects and the broader implications of Bitcoin's growing role in global financial markets. As the cryptocurrency market continues to evolve, advanced modeling techniques like ARMA-EGARCH-WNN-QRNN will be essential in navigating the complex interactions between digital and traditional fiat currencies.

CRediT authorship contribution statement

David Alaminos: Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **M. Belén Salas-Compás:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Manuel Á. Fernández-Gámez:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] I. Abid, E. Bouri, E. Galariotis, K. Guesmi, H. Mzoughi, Bitcoin vs. fiat currencies: Insights from extreme dependence and risk spillover analysis with financial markets, Int. Rev. Financ. Anal. *90* (2023) 102806.
[2] A. Ahmadova, T. Guliyev, K. Aliyev, The Relationship between Bitcoin and Nasdaq, US Dollar Index and Commodities, Int. J. Energy Econ. Policy *14* (1) (2024) 281.
[3] D. Alaminos, M. Guillén-Pujadas, E. Vizuete-Luciano, J.M. Merigó, What is going on with studies on financial speculation? Evidence from a bibliometric analysis, Int. Rev. Econ. Financ. 89 (2024) 429–445.
[4] D. Alaminos, M.B. Salas, M.A. Fernández-Gámez, High-Frequency Trading in Bond Returns: A Comparison Across Alternative Methods and Fixed-Income Markets, Comput. Econ. (2023).
[5] D. Alaminos, M.B. Salas, M.Á. Fernández-Gámez, Hybrid genetic algorithms in agent-based artificial market model for simulating fan tokens trading, Eng. Appl. Artif. Intell. 131 (2024) 107713.
[6] D. Alaminos, M.B. Salas, A. Partal-Ureña, Hybrid ARMA-GARCH-Neural Networks for intraday strategy exploration in high-frequency trading, Pattern Recognit. *148* (2024) 110139.
[7] F. Aliu, A. Asllani, S. Hašková, The impact of bitcoin on gold, the volatility index (VIX), and dollar index (USDX): analysis based on VAR, SVAR, and wavelet coherence, Stud. Econ. Financ. *41* (1) (2024) 64–87.
[8] F. Allen, M.D. Haas, E. Nowak, A. Tengulov, Market efficiency and limits to arbitrage: Evidence from the Volkswagen short squeeze, J. Financ. Econ. *142* (1) (2021) 166–194.
[9] U. Anders, O. Korn, Model selection in neural networks, Neural Netw. 12 (2) (1999) 309–323.
[10] I. Antoniadis, N. Sariannidis, S. Kontsas, The Effect of Bitcoin Prices on Us Dollar Index Price, in: N. Tsounis, A. Vlachvei. (Eds.), Advances in Time Series Data Methods in Applied Economic Research, Springer, Berlin, 2018, pp. 511–521.
[11] A.F. Aysan, N. Isac, O. Drammeh, R. Özcan, Threat of İntervention in Cryptocurrency Market: West Side Story of Bitcoin and Ripple, Econ. Comput. Econ. Cybern. Stud. Res. 57 (4) (2023) 41–56, https://doi.org/10.24818/18423264/57.4.23.03.
[12] S. Barde, Direct comparison of agent-based models of herding in financial markets, J. Econ. Dyn. Control 73 (2016) 329–353.
[13] D.G. Baur, T. Dimpfl, K. Kuck, Bitcoin, gold and the US dollar—A replication and extension. Financ. Res. Lett. 24 (2018) 1–10.
[14] W. Bazán-Palomino, The increased interest in Bitcoin and the immediate and long-term impact of Bitcoin volatility on global stock markets, Econ. Anal. Policy *80* (2023) 1080–1095.
[15] Y. Becerikli, On three intelligent systems: dynamic neural, fuzzy and wavelet networks for training trajectory, Neural Comput. Appl. 13 (2004) 339–351.
[16] Y. Becerikli, Y. Oysal, A.F. Konar, On a dynamic wavelet network and its modeling application, Lect. Notes Comput. Sci.: 2714 (pp (2003) 710–718.
[17] L. Behera, I. Kar, A.C. Elitzur, Recurrent quantum neural network and its applications, Emerg. Phys. Conscious. (2006) 327–350.
[18] M. Benedetti, E. Lloyd, S. Sack, M. Fiorentini, Parameterized quantum circuits as machine learning models, Quantum Sci. Technol. *4* (4) (2019) 043001.
[19] C. Bernard, S. Mallat, J.-J. Slotine, Wavelet interpolation networks, Pap. Presente Proc. (1998) of ESANN'98.
[20] S.A. Billings, H.-L. Wei, A new class of wavelet networks for nonlinear system identification, IEEE Trans. Neural Netw. 16 (4) (2005) 862–874.
[21] I.A. Boboc, M.C. Dinică, An algorithm for testing the efficient market hypothesis, PLoS One *8* (10) (2013) e78177.
[22] Y. Cao, G.G. Guerreschi, A. Aspuru-Guzik, arXiv preprint arXiv:1711.11240, Quantum Neuron.: Elem. Build. Block Mach. Learn. Quantum Comput. (2017).
[23] L. Cao, Y. Hong, H. Fang, G. He, Predicting chaotic time series with wavelet networks, Phys. D. 85 (1995) 225–238.
[24] H.H. Cao, Y. Huang, Y. Huang, B. Yeung, X. Zhang, Fintech, financial inclusion, digital currency, and CBDC, J. Financ. Data Sci. *9* (2023) 100115.
[25] Guangxi Cao, Meijun Ling, Asymmetry and Conduction Direction of the Interdependent Structure between Cryptocurrency and Us Dollar, Renminbi, and Gold Markets, Chaos Solitons Fractals 155 (2022) 111671.
[26] E.I. Cevik, S. Gunay, S. Dibooglu, D.Ç. Yıldırım, The impact of expected and unexpected events on Bitcoin price development: Introduction of futures market and COVID-19, Financ. Res. Lett. *54* (2023) 103768.
[27] A. Chakraborti, I.M. Toke, M. Patriarca, F. Abergel, Econophysics review: II. Agent-based models, Quant. Financ. *11* (7) (2011) 1013–1041.
[28] S.H. Chen, C.L. Chang, Y.R. Du, Agent-based economic models and econometrics, Knowl. Eng. Rev. *27* (2) (2012) 187–219.
[29] S.H. Chen, M. Kaboudan, Y.R. Du (Eds.), The Oxford handbook of computational economics and finance, Oxford University Press, 2018.

[30] Y. Chen, B. Yang, J. Dong, Time-series prediction using a local linear wavelet neural wavelet, Neurocomputing 69 (2006) 449–465.
[31] C. Chiarella, G. Iori, J. Perelló, The impact of heterogeneous trading rules on the limit order book and order flows, J. Econ. Dyn. Control *33* (3) (2009) 525–537.
[32] L. Cocco, G. Concas, M. Marchesi, Using an artificial financial market for studying a cryptocurrency market, J. Econ. Interact. Coord. *12* (2017) 345–365.
[33] L. Cocco, M. Marchesi, Modeling and Simulation of the Economics of Mining in the Bitcoin Market, PloS One *11* (10) (2016) e0164603.
[34] L. Cocco, R. Tonelli, M. Marchesi, An agent-based artificial market model for studying the bitcoin trading, IEEE Access *7* (2019) 42908–42920.
[35] CoinMarketCap. (2021). Bitcoin (BTC) Price, Charts, Market Cap, and other metrics. Retrieved September 24, 2021, from ⟨https://coinmarketcap.com/currencies/bitcoin/⟩.
[36] T. Conlon, S. Corbet, R. McGee, Enduring relief or fleeting respite? Bitcoin as a hedge and safe haven for the US dollar, Ann. Oper. Res. (2024) 1–29.
[37] F.X. Diebold, K. Yilmaz, Measuring financial asset return and volatility spillovers, with application to global equity markets, Econ. J. 119 (2009) 158–171, https://doi.org/10.1111/j.1468-0297.2008.02208.x.
[38] F.X. Diebold, K. Yilmaz, Better to give than to receive: predictive directional measurement of volatility spillovers, Int. J. Forecast. 28 (2012) 57–66.
[39] F.X. Diebold, K. Yılmaz, M. Lynch, P. Gregory, On the network topology of variance decompositions: measuring the connectedness of financial firms, J. Econom. 182 (2014) 119–134, https://doi.org/10.1016/j.jeconom.2014.04.012.
[40] A.H. Dyhrberg, Bitcoin, gold and the dollar—A GARCH volatility analysis, Financ. Res. Lett. 16 (2016) 85–92.
[41] J. Fabus, I. Kremenova, N. Stalmasekova, T. Kvasnicova-Galovicova, An Empirical Examination of Bitcoin's Halving Effects: Assessing Cryptocurrency Sustainability within the Landscape of Financial Technologies. J. Risk Financ. Manag. *17* (6) (2024) 229.
[42] A.G. Felix, S. Jürgen, C. Fred, Learning to forget: continual prediction with LSTM, Neural Comput. 12 (10) (2000) 2451–2471.
[43] L.C. Freeman, A set of measures of centrality based on betweenness, Sociometry (1977), https://doi.org/10.2307/3033543.
[44] L.C. Freeman, Centrality in social networks conceptual clarification, Soc. Netw. 1 (1978) 215–239.
[45] P. Gai, S. Kapadia, Contagion in financial networks, Proc. R. Soc. A Math. Phys. Eng. Sci. 466 (2010) 2401–2423, https://doi.org/10.1098/rspa.2009.0410.
[46] C.-P. Georg, The effect of the interbank network structure on contagion and common shocks, J. Bank. Financ. 37 (2013) 2216–2228.
[47] R. Gibrat, Les Inégalités économiques, Librairie du Recueil Sirey, Paris, France, 1931.
[48] M. Gilli, P. Winker, A global optimization heuristic for estimating agent based models, Comput. Stat. Data Anal. 42 (2003) 299–312.
[49] R. Glick, S. Leduc, The effects of unconventional and conventional US monetary policy on the dollar (May), Fed. Reserve Bank San. Fr. (2013).
[50] C.P.S. Gonçalves, Quantum Neural Machine Learning: Theory and Experiments, Chapter 5. Artificial Intelligence-Applications in Medicine and Biology, IntechOpen, London, 2019, https://doi.org/10.5772/intechopen.84149.
[51] L.K. Grover, Fixed-point quantum search, Phys. Rev. Lett. *95* (15) (2005) 150501.
[52] G.G. Guerreschi, Repeat-until-success circuits with fixed-point oblivious amplitude amplification, Phys. Rev. A *99* (2) (2019) 022306.
[53] H. Halaburda, G. Haeringer, J. Gans, N. Gandal, The microeconomics of cryptocurrencies, J. Econ. Lit. *60* (3) (2022) 971–1013.
[54] M.X. Hanauer, P. Lesnevski, E. Smajlbegovic, Surprise in short interest, J. Financ. Mark. 65 (2023) 100841.
[55] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[56] G.Y. Huang, Y.F. Gau, Z.X. Wu, Price discovery in fiat currency and cryptocurrency markets, Financ. Res. Lett. *47* (2022) 102615.
[57] J. Jagtiani, M. Papaioannou, G. Tsetsekos, E. Dolson, D. Milo, Cryptocurrencies: regulatory perspectives and implications for investors, Palgrave Handb. Technol. Financ. (2021) 161–186.
[58] L. Jiao, J. Pan, Y. Fang, Multiwavelet neural network and its approximation properties, IEEE Trans. Neural Netw. 12 (5) (2001) 1060–1066.
[59] S. Kadambe, P. Srinivasan, Adaptive wavelets for signal classification and compression, Int. J. Electron. Commun. 60 (2006) 45–55.
[60] P. Kayal, P. Rohilla, Bitcoin in the economics and finance literature: a survey, SN Bus. Econ. *1* (7) (2021) 88.
[61] T. Khayamian, A.A. Ensafi, R. Tabaraki, M. Esteki, Principal componentwavelet networks as a new multivariate calibration model, Anal. Lett. 38 (9) (2005) 1447–1489.
[62] B.S. Kim, T.G. Kim, Cooperation of simulation and data model for performance analysis of complex systems, Int. J. Simul. Model. *18* (4) (2019) 608–619.
[63] A. Kirman, Money and Financial Markets, Chapter Epidemics of Opinion and Speculative Bubbles in Financial Markets, Macmillan, New York, USA, 1991, pp. 354–368.
[64] A. Kirman, Ants, rationality, and recruitment, Q. J. Econ. 108 (1) (1993) 137–156.
[65] A. Kraay, Do high interest rates defend currencies during speculative attacks? J. Int. Econ. 59 (2) (2003) 297–321.
[66] P.R. Krugman, A model of balance-of-payments crises, J. Money, Credit Bank. 11 (3) (1979) 311–325.
[67] N. Kunimoto, K. Kakamu, Is Bitcoin really a currency? A viewpoint of a stochastic volatility model, Appl. Econ. 54 (57) (2021) 6536–6550.
[68] S. Lallahem, J. Mania, A. Hani, Y. Najjar, On the use of neural networks to evaluate groundwater levels in fractured media, J. Hydrol. 307 (1–4) (2005) 92–111.
[69] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE *86* (11)) (1998) 2278–2324.
[70] J. Lee, H. Koh, H.J. Choe, Learning to trade in financial time series using high-frequency through wavelet transformation and deep reinforcement learning, Appl. Intell. *51* (8) (2021) 6202–6223.
[71] M. Levy, S. Solomon, New evidence for the power-law distribution of wealth, Phys. A: Stat. Mech. its Appl. *242* (1-2) (1997) 90–94.
[72] M. LiCalzi, P. Pellizzari, Fundamentalists clashing over the book: a study of order-driven stock markets, Quant. Financ. *3* (6) (2003) 470.
[73] Y. Liu, N. Naktnasukanjn, A. Tamprasirt, T. Rattanadamrongaksorn, Comparison of the Asymmetric Relationship between Bitcoin and Gold, Crude Oil, and the US Dollar before and after the COVID-19 Outbreak, J. Risk Financ. Manag. *16* (10) (2023) 455.
[74] G. Liu, F. Xiao, C.T. Lin, Z. Cao, A fuzzy interval time-series energy and financial forecasting model using network-based multiple time-frequency spaces and the induced-ordered weighted averaging aggregation operation, IEEE Trans. Fuzzy Syst. *28* (11) (2020) 2677–2690.
[75] J.W. Liu, F.L. Zuo, Y.X. Guo, T.Y. Li, J.M. Chen, Research on improved wavelet convolutional wavelet neural networks, Appl. Intell. *51* (2021) 4106–4126.
[76] R.P. Mahajan, A Quantum Neural Network Approach for Portfolio Selection, Int. J. Comput. Appl. 29 (4) (2011) 47–54.
[77] S. Martínez-Jaramillo, O.P. Perez, F.A. Embriz, F.L.G. Dey, Systemic, risk, Financ. Contag. Financ. fragility. J. Econ. Dyn. Control 34 (2010) 2358–2374.
[78] R. Matkovskyy, A. Jalan, From financial markets to Bitcoin markets: A fresh look at the contagion effect, Financ. Res. Lett. *31* (2019) 93–97.
[79] A. Mellit, M. Benghamen, S.A. Kalogirou, An adaptive wavelet-network model for forecasting daily total solar-radiation, Appl. Energy 83 (2006) 705–722.
[80] H. Moayedi, M. Raftari, A. Sharifi, W.A.W. Jusoh, A.S.A. Rashid, Optimization of ANFIS with GA and PSO estimating α ratio in driven piles, Eng. Comput. *36* (1) (2020) 227–238.
[81] Khaled Mokni, Ahdi Noomen Ajmi, Cryptocurrencies Vs. Us Dollar: Evidence from Causality in Quantiles Analysis, Econ. Anal. Policy 69 (2021) 238–252.
[82] F. Nawaz, A.F. Aysan, U. Kayani, H. Nasseredine, Do CBDCs promote financial inclusion and strengthen the monetary regulations? J. Infrastruct., Policy Dev. *8* (8) (2024) 5870.
[83] M.A. Nielsen, I.L. Chuang, Quantum computation and quantum information, Phys. Today *54* (2) (2001) 60.
[84] M. Obstfeld, H. Zhou, The Global Dollar Cycle, Brook. Pap. Econ. Act. 2022 (2) (2022) 361–447.
[85] L. Ortega, K. Khashanah, A neuro-wavelet model for the short-term forecasting of high-frequency time series of stock returns. J. Forecast. *33* (2) (2014) 134–146.
[86] Y. Oussar, I. Rivals, L. Presonnaz, G. Dreyfus, Trainning wavelet networks for nonlinear dynamic input output modelling, Neurocomputing 20 (1998) 173–188.
[87] S. Postalcioglu, Y. Becerikli, Wavelet networks for nonlinear system modelling, Neural Comput. Appl. 16 (2007) 434–441.
[88] E. Prasad, TRANSFORMATION OF MONEY: Digital technology is poised to change the very nature of money, MIT Technol. Rev. *125* (3) (2022) 25–32.
[89] L. Qin, N. Yu, D. Zhao, Applying the convolutional neural network deep learning technology to behavioural recognition in intelligent video, Teh. čki Vjesn. *25* (2) (2018) 528–535.
[90] L. Rubio, A. Palacio Pinedo, A. Mejía Castaño, F. Ramos, Forecasting volatility by using wavelet transform, ARIMA and GARCH models, Eurasia Econ. Rev. 13 (3) (2023) 803–830.

[91] V. Shatravin, D. Shashev, S. Shidlovskiy, Sigmoid Activation Implementation for Neural Networks Hardware Accelerators Based on Reconfigurable Computing Environments for Low-Power Intelligent Systems, Appl. Sci. *12* (10) (2022) 5216.
[92] J. Shi, Z. Jincheng, Y. Wang, D.L.K. Chuen, The Evolution and Future of Cryptocurrency-Based Fundraising Mechanisms, J. Br. Block Assoc. (2024).
[93] A.K. Swoboda, The euro-dollar market: An interpretation. The International Monetary System, 1st Edition)., Routledge, London, UK, 1993.
[94] Beata Szetela, Mentel Grzegorz, edek Stanisław G, Dependency Analysis between Bitcoin and Selected Global Currencies, Dyn. Econom. Models 16 (2016) 133–144.
[95] F. Tacchino, C. Macchiavello, D. Gerace, D. Bajoni, An artificial neuron implemented on an actual quantum processor, npj Quantum Inf. *5* (1) (2019) 1–8.
[96] G. Tedeschi, G. Iori, M. Gallegati, Herding effects in order driven markets: The rise and fall of gurus, J. Econ. Behav. Organ. *81* (1) (2012) 82–96.
[97] E. Vasileiou, Does the short squeeze lead to market abnormality and antileverage effect? Evidence from the Gamestop case, J. Econ. Stud. 49 (8) (2022) 1360–1373.
[98] M. Vogl, P.G. Rötzel, S. Homes, Forecasting performance of wavelet neural networks and other neural network topologies: A comparative study based on financial market data sets, Mach. Learn. Appl. *8* (2022) 100302.
[99] K.H. Wan, O. Dahlsten, H. Kristjánsson, R. Gardner, M.S. Kim, Quantum generalisation of feedforward neural networks, NPJ Quantum Inf. 3 (36) (2017), https://doi.org/10.1038/s41534-017-0032-4.
[100] S. Wang, X. Chen, C. Tong, Z. Zhao, Matching synchrosqueezing wavelet transform and application to aeroengine vibration monitoring, IEEE Trans. Instrum. Meas. 66 (2017) 360–372.
[101] L. Wang, P.K. Sarker, E. Bouri, Short-and long-term interactions between Bitcoin and economic variables: Evidence from the US, Comput. Econ. *61* (4) (2023) 1305–1330.
[102] M. Wątorek, S. Drożdż, J. Kwapień, L. Minati, P. Oświęcimka, M. Stanuszek, Multiscale characteristics of the emerging global cryptocurrency market, Phys. Rep. *901* (2021) 1–82.
[103] S. Wu, Co-movement and return spillover: evidence from Bitcoin and traditional assets, SN Bus. Econ. *1* (10) (2021) 122.
[104] J. Xu, D.W.C. Ho, Adaptive wavelet networks for nonlinear system identification. Paper Presented at the Proc. of American Control Conference, San Diego, USA, 1999.
[105] Y. Yang, J. Wang, Forecasting wavelet neural hybrid network with financial ensemble empirical mode decomposition and MCID evaluation, Expert Syst. Appl. *166* (2021) 114097.
[106] G. Yule, A mathematical theory of evolution based on the conclusions of Dr, J. C. Willis, F. R. S.,'' Philos. Trans. B vol. 213 (1925) 21–87.
[107] Q. Zhang, Regress Sel. Wavel. Netw. Constr. (1993).
[108] Q. Zhang, Using Wavel. Netw. Nonparametr. Estim. (No. 2321): Techincal Rep., Inria. (1994).
[109] Q. Zhang, Using wavelet network in nonparametric estimation, IEEE Trans. Neural Netw. 8 (2) (1997) 227–236.
[110] Q. Zhang, A. Benveniste, Wavelet networks, IEEE Trans. Neural Netw. 3 (6) (1992) 889–898.
[111] J. Zhao, B. Chen, J. Shen, Multidimensional non-orthogonal waveletsigmoid basis function neural network for dynamic process fault diagnosis, Comput. Chem. Eng. 23 (1998) 83–92.
[112] M. Zidan, A.-H. Abdel-Aty, M. El-shafei, M. Feraig, Y. Al-Sbou, H. Eleuch, M. Abdel-Aty, Quantum Classification Algorithm Based on Competitive Learning Neural Network and Entanglement Measure, Appl. Sci. 9 (2019) 1277.
[113] M. Zolfaghari, S. Gholami, A hybrid approach of adaptive wavelet transform, long short-term memory and ARIMA-GARCH family models for the stock index prediction, Expert Syst. Appl. 182 (2021) 115149.