Computational Economics
https://doi.org/10.1007/s10614-025-11026-8

High-Frequency Trading, Short Squeeze and ARMA-GARCH-Fractal Neural Networks

David Alaminos[1] · **M. Belén Salas-Compás**[2] · **Estefanía Alaminos**[3]

Accepted: 9 June 2025

Abstract
In recent years, short squeeze events, such as the GameStop case in early 2021, have gained prominence, highlighting the need for advanced analyses of such phenomena. While traditional econometric and neural network approaches have struggled with predictive accuracy, our study addresses these gaps by analyzing the GameStop short squeeze using high-frequency intraday market data. We propose a novel hybrid approach that integrates an Autoregressive Moving Average-Generalized Autoregressive Conditional Heteroscedasticity model with Neural Networks, as well as exploiting fractal dynamics to capture multiscale temporal dependencies and hierarchical patterns in financial markets. This fractal framework effectively addresses the nonlinear and chaotic dynamics of the financial markets. Our methods deliver high predictive accuracy, with the ARMA-GARCH-Quantum approach standing out. This method highlights its greater adaptability and accuracy, proving the benefits of integrating fractal principles into predictive modeling. By enhancing adaptability and precision, this study contributes valuable tools for market forecasting and risk management, aiding regulators and financial managers in monitoring and mitigating abnormal price movements that could distort markets or spark crises.

Keywords High-Frequency Trading · Short Squeeze · ARMA-GARCH · Fractal Neural Networks · Deep Learning · Quantum Computing

JEL Classification C45 · C53 · G01 · G12 · G14

✉ David Alaminos
alaminos@ub.edu

[1] Department of Business, University of Barcelona, Av. Diagonal 690, 08034 Barcelona, Spain

[2] Department of Finance and Accounting, University of Málaga, Campus de El Ejido 6, 29071 Málaga, Spain

[3] Chair of Sustainable Economics and Finance, University of Málaga, Campus de El Ejido 6, 29071 Málaga, Spain

Published online: 05 August 2025

1 Introduction

Stock market analysis is always a topic of interest to researchers who aim to demonstrate that financial performance can be examined in advance. However, financial events are well-known and difficult to predict. Traders and investors pursue competitive benefits through the analysis of market data like press reports, company statements, company forecasts, industry expectations, technical tendencies, and trading behaviour to identify helpful negotiation signals in anticipation of expected price movements (Bui & Ślepaczuk, 2022; Godfrey, 2016). Since the beginning of the coronavirus pandemic, corporate closures to stop the propagation of the virus have led to a greater interest in the stock market by individual investors, owing to the availability of more free time, capital, and commission-free stockbrokers. Such interest led to the short squeeze event of January 2021, which was largely driven by the organised trading movements of the r/WallStreetBets subreddit, whose user base has expanded quickly since the episode (Andreev et al., 2022). A short squeeze involves an uncommon situation that causes a fast increase in the price of a stock. This happens when a stock contains a substantial number of short sellers, implying a large number of investors placing bets that the price of the stock will fall. Short squeeze starts when the price rises suddenly and short sellers choose to reduce losses and exit their positions (Feinstein, 2022). In the current financial environment, where retail and non-professional investors are increasingly influential given the appearance of commission-free trading and leverage platforms, the short squeeze event seems highly unlikely to be isolated (Mancini et al., 2022).

Identifying short squeezes involves understanding the process by which they occur. According to Jiang et al. (2021), short squeezes are caused by"pressure on short sellers to cover their positions as a result of sharp price rises or difficulty in borrowing the security in the sellers short". The definition identifies two necessary conditions for a short squeeze to happen: a sharp rise in the share price, requiring short sellers to cover their short positions, and the price rise is accompanied by a short selling constraint, preventing new short sellers from entering the market and arbitraging the elimination of the mispricing.

The most notable instances of short squeezes in the stock markets were seen with Volkswagen and Gamestop. In October 2008, Volkswagen stood as one of the most valuable companies globally. On 26 October 2008, Porsche, a competing car manufacturer that had consistently maintained a minority stake in Volkswagen, made a surprising announcement, disclosing that it had gained control of 74% of Volkswagen's voting shares by acquiring almost all of the company's outstanding shares (Godfrey, 2016). Given the prevailing global financial crisis at the time, short investments were spiraling out of control. The short squeeze from Porsche onto Volkswagen was only feasible because a significant portion of the latter's shares (approximately 12.5%) had been lent to short investors at the time of Porsche's announcement. When the market opened the following day, these investors hurried to close their positions to minimise their losses, purchasing more shares and further inflating the share price. The price of Volkswagen stock surged from around €200 to €1000 (Allen et al., 2021). The Volkswagen squeeze held immense economic significance. According to Godfrey (2016),

hedge funds might have incurred losses of up to 30 billion euros when their short-selling strategy for Volkswagen shares went awry.

In early 2021, GameStop (GME) stock surged, driven by small investors organized on Reddit, particularly on the r/WallStreetBets subreddit, who, despite the company's weak fundamentals, collaborated to take long positions and trigger a short squeeze against large hedge funds (Chohan, 2021; Vasileiou et al., 2021; Andreev et al., 2022). This phenomenon involved three key players: GME, short sellers, and buyers organized on Reddit (Haq et al., 2022; Xiang & Dabbagh, 2022). This case highlights the role of social media as a powerful tool in financial markets, enabling effective collective action by retail investors for the first time (Choy et al., 2021). r/WSB users coordinated their actions to drive up stock prices, generating dynamic networks of interaction (Zheng et al., 2021).

In parallel, tools such as Google searches and posts on platforms like Twitter have been leveraged to analyze financial behaviors through sentiment analysis, although this approach presents methodological challenges (Bangyal et al., 2021; Qasim et al., 2022). Since individual investors do not always act rationally, there has been a search for strategies to optimize market movements to accurately predict them. In the case of GME, the timing and conditions of the short squeeze turned irrational behavior into profitable behavior (Grachev, 2017). Therefore, accurate stock price prediction is crucial to maximize returns and even anticipate events such as short squeezes (Arashi & Rounaghi, 2022). Although Andreev et al. (2022) attempted to predict trading signals using WSB data and popular stock prices, their models failed to establish profitable strategies, suggesting that future research could improve methods for greater accuracy..

In earlier research, a substantial portion of techniques employed in technical analysis to predict stock market trends traditionally relied on statistical data, despite the stock market's nature as a nonlinear and unpredictable system influenced by political, economic, and psychological factors. Recently, advancements in data-driven Machine Learning (ML) methods have significantly enhanced the modeling of financial market dynamics. ML is gaining prominence in the field of Finance due to its efficacy in managing intricate decision-making processes and mitigating high risks (Sokolovsky & Arnaboldi, 2020). Various researchers have pioneered hybrid models. Tseng et al. (2002) integrate forecast results derived from Autoregressive Integrated Moving Average (ARIMA) and residuals as inputs to Neural Networks (NN) to predict two seasonal time series related to the total output value in Taiwan's machinery industry. The results indicated that the hybrid model outperformed both ARIMA and NN. Khashei and Bijari (2012) propose hybrid approaches that involved incorporating ARIMA residuals and original data as inputs to NNs, thereby enhancing the accuracy of ARIMA in predicting weekly GBP/USD exchange rates.

In recent years, two prominent non-linear models that have demonstrated robust performance in financial markets are the ARIMA model and the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model. While the ARIMA model primarily concentrates on predicting the conditional mean of future values, the heightened market volatility driven by factors such as economic or political instability, as well as global events like pandemics or wars, emphasizes the need for models capable of concurrently predicting both the conditional mean and the conditional

heteroscedasticity of the process. This necessity has led to the adoption of the Autoregressive Moving Average-Generalized Autoregressive Conditional Heteroskedasticity (ARMA-GARCH) model (Emenogu et al., 2019; Ghani et al., 2019). In a study examining NASDAQ's daily stock market index from 2000 to the end of 2016, Arashi and Rounaghi (2022) employ the ARMA-GARCH model and demonstrated its excellent forecasting capabilities, achieving a 1% error level. Their findings also unveiled a correlation between stock price indexes across different time scales, portraying the NASDAQ stock exchange as an efficient and non-fractal market. To advance research in this domain, they proposed exploring hybrid models based on neural networks or genetic algorithms for predicting chaotic series. Additionally, Hu et al. (2020) conclude by suggesting future research avenues to investigate whether combining the ARMA-GARCH model with the ARMA and GARCH patterns produces optimal fitting effects and prediction abilities.

To address the gap in the literature, our research aims to analyze the short squeeze event from January 11, 2021, to December 31, 2021, using intraday market data and a novel approach that integrates fractal properties into advanced machine learning (ML) methodologies. Specifically, we apply the ARMA-GARCH process combined with fractal-enhanced neural network frameworks, including Neural Networks, Fractal Deep Recurrent Convolutional Neural Networks, Fractal Neural Decision Trees, Fractal Quantum Neural Networks, and Fractal Quantum Recurrent Neural Networks. These fractal adaptations enable hierarchical and multiscale modeling, enhancing the capacity to capture the complex dynamics of financial markets. For comparison, we also utilize the ARMA-GARCH-Neural Network method as referenced by Sun et al. (2019).

The short squeeze of certain highly speculative stocks represents a crucial event that likely surprised many market participants, highlighting the need for a detailed examination of this phenomenon Our study delves into the short squeeze's impact on stock market quality and price formation, offering insights that are especially relevant given the increasing influence of retail and non-professional investors facilitated by trading platforms. We consider this analysis to be a significant contribution to literature, as the event is unlikely to remain isolated in the current financial context. The rapid growth of communities like WallStreetBets (WSB) suggests the potential for future events to generate similarly abnormal returns, underscoring the practical value of our research for future investors.

Additionally, our study diverges from traditional statistical and econometric approaches by employing advanced machine learning methods with fractal properties. These ML techniques are highly effective in discerning patterns, correlations, and anomalies within extensive and complex datasets. By integrating fractal dynamics, these methods further enhance adaptability and robustness in capturing market behaviors. This adaptability is particularly critical in the dynamic and ever-evolving financial environment, enabling better responses to systemic changes and mitigating the effects of"momentary irrationality."This research demonstrates the potential of fractal-inspired neural network methodologies to address the complexities of modern financial markets and contribute to their broader understanding (Ferreira et al., 2021; Han et al., 2021; Hansen, 2020).

The remainder of the paper is structured as follows. Section 2 provides an in-depth review of the existing literature. In Sect. 3, the methodology is outlined. Section 4 elaborates on the model, sample, and data used in the research. Section 5 presents the results and findings. Section 6 engages in a discussion of the obtained results. Finally, Sect. 7 concludes by explaining the reached conclusions.

2 Literature Review

Existing literature on short squeeze has four main lines of research. Firstly, the approach focused on the specific case of GME (January 2021), which drew the attention of academics, practitioners, and regulators. Secondly, the role of social networks (especially Reddit). Thirdly, the behavior of digital communities (WSB). Finally, the methodology used by various authors to improve predictive capacity.

Regarding the first line of research, one of the relevant approaches has focused on identifying the determining factors of the event, including variables such as market volume, the put-call ratio, and the interest reflected in Google searches (Hilliard & Hilliard, 2023; Umar et al., 2021; Vasileiou, 2021). Umar et al. (2021) examine indicators including the count of news publications, put-call ratio, and short-sale volume. They apply an econometric model to daily data spanning from 01 January 2020 to 30 January 2021. Their findings suggest that the put-call ratio positively influences GME returns before the peak of the GME saga, playing a role in the surge of prices in January. Additionally, they identify a positive association between GME returns and short sales volume during the GME episode, confirming the occurrence of the short squeeze event. Moreover, they reveal a negative correlation between the count of news posts and GME's stock price. This implies that the sentiments of investors averse to media may not have a positive impact on GME's returns during this period. Consequently, this suggests that this segment of investors may not be blamed behind the recently escalating rivalry between Reddit's social media investors and the hedge funds involved. Vasileiou (2021) utilises a Google Trends index incorporating terms related to the GME case, employing intraday hourly data covering the period from 4/1/2021 to 26/3/2021. This author shows the existence of two-way linkage causality from GME trading volume to GME outcomes, along with a notable single causality from Google queries to GME yields. Additionally, they employ various GARCH models, such as EGARCH, TGARCH, and IGARCH. Their conclusion suggests that the GARCH (1, 1) model, incorporating a t-distribution in the error term, proves most suitable for their dataset. This model effectively addresses autocorrelation and ARCH-LM issues and exhibits the lowest values for both the Akaike and Schwarz Information Criteria among the considered models. Hilliard and Hilliard (2023) employ the put-call parity (PCP) concept to explore potential violations of the no-arbitrage condition in the context of the GME stock squeeze. The research covers three distinct time periods: the period before the stock squeeze (from January 4 to January 21), the squeeze period (from January 22 to February 10), and the period after the squeeze (from February 11 to February 26). The points marking the boundaries of these periods were determined based on significant changes in both trading volumes and stock prices. The authors identify instances of breaches in the no-arbitrage

condition when Positive Cash Portfolio (PCP) strategies generate positive returns, accounting for bid-ask spreads. Notably, a higher frequency of breaches is noticeable in both the pre-squeeze and squeeze times. The most significant instances of breaches are observed in the put-call parity (PCP) related to selling or shorting stocks. Furthermore, breaches are more prominent in the context of longer maturities. The authors acknowledge several obstacles related to short selling, such as failures to deliver, stock borrowing rates, and stock availability. Through regression models, they determine that these obstacles significantly contribute to deviations from PCP predictions. Their primary finding is that, during the GME squeeze, the market predominantly adhered to rationality, considering no-arbitrage conditions adjusted for frictions.

Concerning the role of social networks, Long et al. (2021) investigate the impact of discussions on the r/WallStreetBets (WSB) subreddit on GME's price dynamics, contributing to the literature on media sentiment. They create a Reddit-specific investment lexicon, assessing it with intraday data from GameStop using both lexical and machine learning approaches. Analyzing high-frequency stock price data for GME from January 1, 2021, to February 28, 2021, the lexical approach connects words to sentiments, while the machine learning approach predicts sentiment of new input text based on previously labeled data. Their findings caution individual investors that social media discussions may not reliably protect investments. Mancini et al. (2022) scrutinize WSB community discussions from September 1, 2019, to February 1, 2021, using the Valence Aware Dictionary and Sentiment Reasoner (VADER) algorithm. Findings reveal a shift from diverse to uniform opinions as user participation intensifies, qualitatively linked to the sharp rise in GME's stock price. Andreev et al. (2022) explore the consequences of the January 2021 short squeeze, utilizing sentiment data from WSB. They combine this with fundamental and technical indicators to train a machine learning model capable of forecasting stock performance one, five, and seven days ahead. The most effective model, a Random Forest autoregressive model, forecasts the stock's position seven days ahead with an estimated accuracy rate of 70 percent, providing insights into self-organized collective actions observed on social networks.

Jarrow and Li (2021) investigate the impact of coalitional chatroom operators on stock prices through shock trading across various social media platforms. They examine the economic implications of the coalition's formation within a micro-founded quasi-competitive equilibrium framework that incorporates strategic trading. In this scenario, a significant trader strategically opts for short selling of a specific stock, possessing a comprehensive understanding of the expected influence on the stock's price due to their trade. The study reveals that media groups can influence and discipline the incentive of large traders to engage in short selling, but this influence may vary in terms of allocative efficiency. Additionally, they demonstrate that, when employing a belief-neutral welfare criterion, media groups discipline the large operator's short-selling incentive, leading to uniformly improved welfare outcomes. Paliewicz (2023) investigates the language and persuasive strategies employed in the context of social and economic protest linked to the 2021 short squeeze. This movement, influenced by populist rhetoric found on the subreddit forum r/WallStreetBets, led to a short squeeze with significant financial implications for hedge funds, resulting in substantial losses. The author's argument centres on the notion that investors were swayed

and spurred into action by market-driven entities employing emotions such as anger and nostalgia, particularly during the period of the COVID-19 pandemic when many individuals felt a sense of helplessness. The conclusion drawn is that, while engagement in the market carries tangible financial risks, including a rhetorical panic, it also underscores how investors can utilise a form of market-based subversion, known as détournement, to their advantage. Finally, Vaughan et al. (2023) explore the GameStop short squeeze across three distinct online spaces: the WallStreetBets subreddit, the #GameStop hashtag on Twitter, and pertinent digital news sources in the United States. Their aim is to interpret the observed patterns in these outcomes through the concept of"connective action."The authors utilise a combination of non-negative matrix factorization (NMF) thematic models and manual content analysis techniques. Their findings lead to the conclusion that digital platforms possess the ability to establish boundaries and points of intermediation within the realm of contentious politics. They particularly concentrate on the ways in which action repertoires, collective identities, and discourses are shaped. These established boundaries serve to restrict both discursive and technological connections, effectively segregating users into relatively isolated segments and, consequently, diminishing their influence and impact on broader discursive systems.

Several research studies (Haq et al., 2022; Kim et al., 2023; Vasileiou, 2021; Zheng et al., 2021) have focused on analyzing the behavior of the WSB community during the GME short squeeze event through. In Zheng et al. (2021), dynamic interaction networks are constructed to investigate the collective behaviors of users engaged in GME discussions on the WSB subreddit. Utilizing the latent Dirichlet allocation (LDA) statistical model, the study reveals concealed semantic structures within the text and highlights greater central shifts in debated topics, accompanied by stronger and more varied participant sentiments. Additionally, the research discloses that a portion of GME's stock price is influenced by social network activity, the prevalence of predominant topics, and sentiment divergence among users on r/WSB. Haq et al. (2022) examine the conduct of newcomers on the WSB subreddit and their reception by the existing community. The study explores the correlation between traditional stock market prices and online activity, comparing the behavior of newcomers with seasoned users. Topic analysis is employed to assess how community feedback influences user categories over different time periods, revealing significant differences in communication styles, with newcomers adopting a distinctive linguistic style and specific emojis characteristic of the WSB community. Kim et al. (2023) explore investor conduct on social media platforms during the GME short squeeze, formulating hypotheses related to Reddit, Twitter, and GameStop-related stock trading data. The study utilizes statistical methods, including linear regression, to scrutinize the data, revealing that individual investors largely adhered to trading patterns posited by the authors on social media platforms. Social sentiment played a role in reinforcing collective behavior, with Reddit's social information having a more pronounced effect on GME stock trading during the short squeeze compared to Twitter. Bangyal et al. (2021) utilize eight machine-learning algorithms to discern sentiments in COVID-19-related fake news, demonstrating a noteworthy level of accuracy compared to alternative models. Qasim et al. (2022) apply nine transfer learning models for binary text classification on datasets related to COVID-19 fake

news, COVID-19 English tweets, and an extremist-non-extremist dataset. Finally, Vasileiou et al. (2021) examine the GME short squeeze using intraday data, reporting empirical support for anomalous behavior in GME's stock price. The findings suggest non-random distribution of GME yields over the sample period, with GARCH models indicating that volatility rose with higher GME prices, contrary to typical time-series behaviour.

Lastly, over the past few years, novel methods like neural networks and fuzzy neural network approaches have been utilised for various forecast challenges across financial, economic, and commercial domains. This trend is evident in the work of Jiang et al. (2021), who assert that their study results indicate the inadequacy of the ordinary least squares regression method in effectively predicting the stock market. They undertake multivariate analyses to validate their strategy for identifying the appearance and size of a short squeeze.

Several researchers have proposed hybrid models to enhance forecasting accuracy. In one study, Tseng et al. (2002) integrate prediction results from Autoregressive Integrated Moving Average (ARIMA) and residuals as inputs for Neural Networks (NN) to predict seasonal time series related to Taiwan's machinery industry output. Their findings demonstrated the superior performance of the hybrid model over both ARIMA and NN models. Another approach, presented by Khashei and Bijari (2012), introduce hybrid methods utilizing ARIMA residuals and original data as inputs for NNs, resulting in improved predictions for weekly GBP/USD exchange rates compared to ARIMA alone. Hu et al. (2020) suggest future research exploring the optimization of combining the ARMA-GARCH model with ARMA and GARCH models for fitting and predictive performance. In a study spanning from 2000 to 2016 on the NASDAQ stock exchange, Arashi and Rounaghi (2022) utilize the ARMA-GARCH model, achieving highly accurate forecasts with a 1% error level. They identified a correlation among stock price indexes across different time scales, suggesting the NASDAQ operates as an efficient and non-fractal market. Proposing future avenues, they advocated for applying hybrid models based on neural networks or genetic algorithms for predicting chaotic series. Sun et al. (2019) introduce a machine learning methodology named ARMA-GARCH-NN to understand patterns within daily stock market shocks for forecasting. This innovative approach integrates traditional financial pricing models with artificial neural networks, employing well-structured feature selection and cross-validation techniques. Empirical assessments using high-frequency data from the USA stock market indicate the method's potential to forecast market shocks and validate its proficiency in identifying patterns within extensive stock market data without the need for strong assumptions about data distribution.

Continuing with the mixed methods, Paquet and Soleymani (2022) introduce a pioneering hybrid deep quantum neural network known as the"QuantumLeap"system, designed for financial predictions. The QuantumLeap system are formed with three primary elements: an encoder that converts segmented financial time series data into a sequence of density matrices, a deep quantum network tasked with forecasting the density matrix at a subsequent point in time, and a classical network that computes the maximum price attained by a security at that future time using the output density matrix. Through their experimental findings involving 24 different securities, the authors compellingly demonstrate the system's precision and efficiency, especially

in scenarios involving both regression and extrapolation. In the research conducted by Liu and Long (2020), they introduce an enhanced hybrid forecasting framework for predicting and analysing daily stock closing prices in the USA and China across different time intervals. To validate its efficacy, their hybrid model is compared with several benchmark models. The investigation reveals that the proposed hybrid framework outperforms other deep learning methods and individual models in terms of prediction accuracy. The forecasting outcomes have valuable applications in stock market monitoring and financial data analysis. The study conducted by Wu et al. (2022) focuses on the analysis of time series graphs derived from market price data. They introduce an innovative framework designed to address fundamental questions by utilising structural information extracted from these price graphs. By incorporating deep learning models alongside this structural information, the study achieves robust performance and demonstrates practical utility in the domains of stock prediction and trading. The efficacy of its novel approach is evidenced by the use of real stock market data, and its framework outperforms a number of contemporary benchmarks. Moreover, in trading simulations, their framework yields the largest aggregate returns. These results broaden the current financial sector applications of complex network methodologies and offer valuable insights for the practice of investment, particularly in the context of decision support in the financial market.

Recent studies, such as the work conducted by Syuhada et al. (2023), have embraced hybrid models, undertaking a comparative analysis between the GA-based ARMA-GARCH model and various Artificial Intelligence (AI) models. This study evaluates linear regression, support vector regression, multilayer perceptron, and long short-term memory in forecasting Value at Risk (VaR), utilizing Bitcoin, crude oil, and stock index returns. The findings highlight the potential enhancement in VaR forecasting through the synergy of these models. The authors recommend further exploration of GA optimization alongside other time series models, encompassing threshold GARCH (TGARCH), exponential GARCH (EGARCH), asymmetric power ARCH (APARCH), and their fractionally integrated variations (e.g., FIGARCH, FIEGARCH, and FIAPARCH), which can capture more intricate empirical characteristics of asset returns and volatility. In another study from 2023, Wang et al. (2023) introduce the"Primary Ensemble Empirical Mode Decomposition combined with Quantum Neural Network"(PEEMD-QNN) model for forecasting Chinese stock index time series. This innovative model outperforms other methods in stock market prediction, including the Back Propagation Neural Network, QNN model, and Ensemble Empirical Mode-Quantum Neural Network (EMD-QNN) model. Additionally, Cao et al. (2023) propose a hybrid quantum computing framework for carbon price forecasting, utilizing the Quantum Long Short-Term Memory (QLSTM) model. Their model, Linear-layer-enhanced Quantum Long Short-Term Memory (L-QLSTM), incorporates linear layers to extract features, reduce quantum bits, and amplify quantum advantages. Testing with real-world data from the European Union Emission Trading (2017–2020) shows that the suggested L-QLSTM approach significantly enhances learning accuracy compared to the standard QLSTM method.

As a summary, Table 1 provides a comparison of the methods used in previous literature.

Table 1 Methods used in studies on financial markets prediction

Author	Year	Method	Main topic
Arashi and Rounaghi	2022	ARMA-GARCH-NN	NASDAQ stock exchange
Cao et al.	2023	QLSTM	Carbon price forecasting
Liu and Long	2020	Hybrid forecasting framework	Time-interval analysis of U.S. and China daily closing prices
Paquet and Soleymani	2022	QuantumLeap system	Financial predictions
Sun et al	2019	ARMA-GARCH-NN	Daily shocks in stock market forecasting
Syuhada et al	2023	GA-based ARMA-GARCH	VaR forecasting using Bitcoin, oil, and index returns
Tseng et al	2002	ARIMA-NN	Forecast Taiwan's machinery output trends
Wang et al	2023	PEEMD-QNN	Chinese stock index forecasting
Wu et al	2022	Deep learning models	Time series analysis of market prices

In summary, the results of previous research indicate that econometric methods have not been sufficiently successful in predicting the stock market. Our research covers the lack of short squeeze event analysis using intraday stock market data and the absence of studies combining ARMA-GARCH models with machine learning. Recently, NN methods have been used for various types of forecasting problems in different sciences, including finance, to improve stock price predictions. We can conclude that no previous literature has approached the analysis of the short squeeze event on intraday market data with an ARMA-GARCH process using different ML methods, as well as making a comparison between them.

3 Methodology

We introduce a novel modeling approach that integrates ARMA-GARCH with Neural Networks, enhanced by Deep Learning and Quantum machine learning methodologies with fractal properties. Specifically, we explore advanced techniques such as ARMA-GARCH-NN, ARMA-GARCH-DRCNN, ARMA-GARCH-DNDT, ARMA-GARCH-QNN, and ARMA-GARCH-QRNN. These methods leverage fractal dynamics to capture multiscale temporal dependencies and hierarchical patterns in financial markets. The overall framework is illustrated in Fig. 1 and consists of three primary components: estimating market shocks, extracting and selecting relevant features, and optimizing the model.

The process begins by calibrating an ARMA-GARCH model on high-frequency equity yields to detect market disruptions and quantify volatility dynamics. Next, we perform feature extraction and selection, incorporating fractal properties, to identify the most significant variables for forecasting market disruptions. Finally, artificial neural networks enhanced with fractal scaling and joint learning techniques are employed to predict future market shocks with greater precision.

The ARMA-GARCH-NN method, previously utilized by Sun et al. (2019), serves as a baseline for comparison under our datasets and conditions. Additionally, we introduce two prior methodologies, MS-ARMA-GARCH-MLP and MS-ARMA-GARCH-RBF, to benchmark their performance against traditional approaches. This

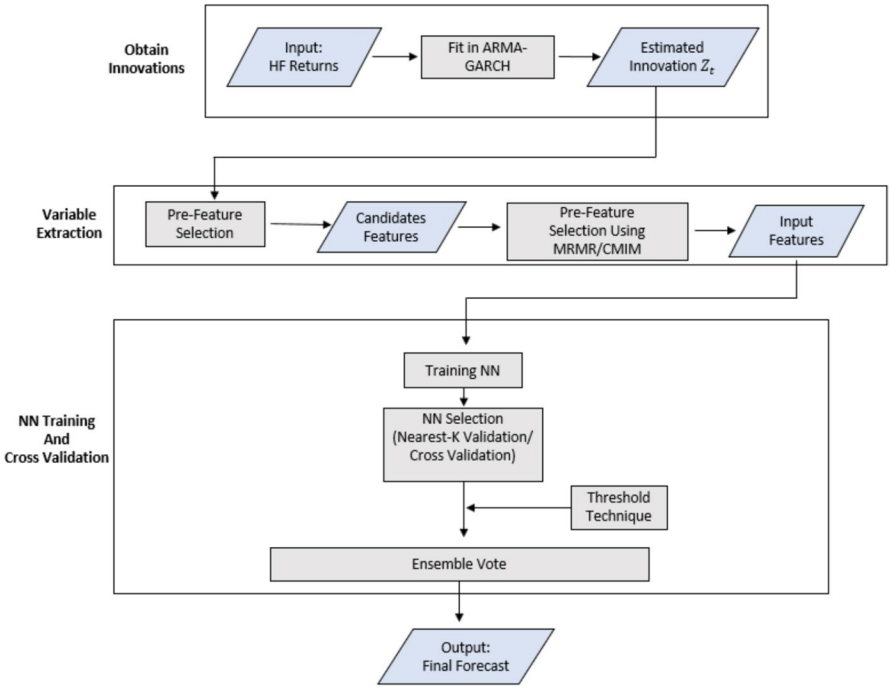

Fig. 1 The ARMA-GARCH-Machine Learning Methods Approach

fractal-informed framework offers a robust and innovative way to model complex financial dynamics and improve predictive accuracy in volatile markets.

3.1 ARMA-GARCH Model

3.1.1 Stock Market Estimation

We calculate the stock market applying the traditional ARMA-GARCH model, which has been proven to be successful in numerous research studies on financial prices (Akgiray, 1989; Jacob, 2015). Given that r_t represents the market yield in period t, the general form of the ARMA (p, q)-GARCH(m, s) model is as follows:

$$r_t = c + \sum_{i=1}^{p} \varphi_i r_{t-i} + \varepsilon_t + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} \tag{1}$$

$$\varepsilon_t = \sigma_t z_t, z_t \sim N(0,1) \tag{2}$$

$$\sigma_t^2 = \omega + \sum_{k=1}^{m} \alpha_k \varepsilon_{t-k}^2 + \sum_{l=1}^{s} \beta_l \sigma_{t-l}^2 \tag{3}$$

Equation (1) outlines the ARMA component, while Eqs. (2) and (3) elaborate on the GARCH component. The terms $r_{t-i}, \ldots, r_{t-p}$ represent autoregressive elements, denoting historical market returns over the last p periods. Similarly, $\varepsilon_{t-1}, \ldots, \varepsilon_{t-q}$ denote moving mean terms that measure the lagged q errors. The coefficients of the model, namely φ_i,θ_j,α_k, and β_l capture varios aspects. The constants c and ω pertain to the model´s constant terms.

The GARCH component captures the autocorrelation property of ε_t, reflecting the clustering characteristics of volatility in stock returns. To elaborate, ε_t is expressed as the product of σ_t and z_t (Eq. (2)), where z_t is the innovation term in the ARMA-GARCH model. It is important to note that the lag orders (i.e., p, q, s, and m) may be determined based on the model's goodness-of-fit, although they are typically set to one. Researchers commonly favour an ARMA (1,1)-GARCH (1,1) model when analysing high-frequency financial data (Jacob, 2015). Consequently, the calculation of market shocks using the ARMA-GARCH model can be articulated as follows:

$$z_t = \frac{r_t - \left(\widehat{c} + \widehat{\varphi_1} t_{t-1} + \widehat{\theta_1}\varepsilon_{t-1}\right)}{\sqrt{\widehat{\omega} + \widehat{\alpha_1}\epsilon_{t-1}^2 + \widehat{\beta_1}\sigma_{t-1}^2}} \tag{4}$$

3.2 Feature Selection

This phase in the methodological process entails collecting a set of explanatory variables to act as inputs for the forecasting models. Initially, we establish a pool of potential features comprising historical time series components of the ARMA-GARCH model. Furthermore, we include data related to market microstructure, covering high frequency technical indicators and volatile measures. The aim is to extract characteristiques out of the past records.

We select the most effective set of features by assessing the mutual information according to Shannon's principle of machine learning (Shannon, 2001). The mutual information between two variables, X and Y, can be described as follows:

$$\boldsymbol{I}(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{\boldsymbol{N}} \sum_{\boldsymbol{i}=1}^{\boldsymbol{N}} \boldsymbol{ln} \frac{\boldsymbol{P}_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{X}_{\boldsymbol{i}},\boldsymbol{Y}_{\boldsymbol{i}})}{\boldsymbol{P}_{\boldsymbol{X}}(\boldsymbol{X}_{\boldsymbol{i}})\,\boldsymbol{P}_{\boldsymbol{Y}}(\boldsymbol{Y}_{\boldsymbol{i}})} \tag{5}$$

where $\boldsymbol{P}_{\boldsymbol{X}}(\boldsymbol{X}_{\boldsymbol{i}})$, $\boldsymbol{P}_{\boldsymbol{Y}}(\boldsymbol{Y}_{\boldsymbol{i}})$, and $\boldsymbol{P}_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{X}_{\boldsymbol{i}},\boldsymbol{Y}_{\boldsymbol{i}})$ represents the marginal and joint density functions from the sample data. It is crucial to highlight that this type of mutual information is nonparametric and does not depend on simplifying assumptions. In a more intuitive sense, mutual information indicates the average reduction in surprise between information sources (e.g., a feature X and the target z), with higher mutual information indicating more accurate predictions. In addition, the redundancy of a chosen group of characteristics is computed in the following way (Ling & McAleer, 2003):

$$\boldsymbol{RD}(\boldsymbol{S}) = \frac{1}{|\boldsymbol{S}|^2} \sum_{\boldsymbol{X}_i, \boldsymbol{X}_j \in \boldsymbol{S}} \boldsymbol{I}(\boldsymbol{X}_i, \boldsymbol{X}_j) \tag{6}$$

where S represents the set of features. Utilising redundancy and mutual learning, we establish method of feature selection in two steps, comprising an initial selection procedure and then primary screening process.

Involving pre-selecting noise filtering, the initial stage is crucial. Given the substantial number of potential features our process deals with, the primary aim in this pre-selection phase is to efficiently eliminate features with limited relevance to the market shock. For this purpose, we create a time series of white noise, denoted as $\boldsymbol{u_t}$ and assess its significance. If a characteristic proves less influential for the target compared to white noise, we conclude it provides inadequate information and should be excluded. To bolster the reliability of the comparison, we repeat the generation of random $\boldsymbol{u_t}$ multiple times and rely on the mean mutual information between $\boldsymbol{u_t}$ and the market shock $\boldsymbol{z_t}$. The pre-selection algorithm eliminates undesirable applicants and reduces the computational cost for the subsequent feature selection round to a minimum.

The subsequent phase involves the curation of pertinent features. Battiti (1994) characterises feature selection as the process of pinpointing the optimum and most noteworthy features within a given feature set. In this investigation, we deploy two adaptations of Battiti's methodology, particularly addressing aspects of relevance and redundancy. Initially, we employ the Forward Selection Minimal-Redundancy-Maximal-Relevance (FSMRMR) criterion introduced by Meyer et al. (2008). This criterion utilises the mean bivariate mutual information as a surrogate in the feature selection procedure. The ensemble of chosen variables, denoted as $\boldsymbol{X_{input}}$, undergoes continual refinement by FSMRMR, substituting the subsequent variable that optimally balances mutual information and redundancy. Secondly, we adopt an alternative approach known as Conditional Mutual Information Maximisation (CMIM) (Fleuret, 2004). This strategy places greater emphasis on redundancy, with the intention of scrutinising the significance of both relevance and redundancy in addressing our predictive challenge. Considering that MRMR maintains equilibrium between heightened relevance and diminished redundancy simultaneously throughout the screening procedure, it may potentially yield superior result as applied to our specific issue.

3.3 Ensemble Neural Networks

Considering the market schoks (z) and the distinctive features (Ω), by optimising the forecasting function g(.) we obtain the equation (Sun et al., 2019):

$$z_t \sim g(r, \varepsilon, \sigma, \Omega \,|t-1) \tag{7}$$

The rationale behind the modelling of the g(.) function is to strike a balance between under-fitting and over-fitting. An under-fitting function lacks optimality in terms of prediction accuracy, while an over-fitting function proves ineffective in predicting out-of-sample events, such as future market shocks. Neural Networks (NNs) dem-

onstrate the ability to capture intricate data dependencies, but the challenge lies in avoiding overfitting. To address this, we employ ensemble NNs in our forecasting approach, as proposed by Hussain et al. (2022).

Particularly, we segment the historical data into a training set and a validation set for each mobile window. Afterward, V Neural Network (NN) models undergo training using the bootstrapping method. Only the models selected for further estimation are those displaying satisfactory performance (the top v% of models) on the validation set. Due to the intrinsic temporal sensitivity in our time series problem, we adopt a novel cross-validation approach known as nearest K cross-validation (NK-CV) to assess the output of each NN model. Assuming the sample size of each moving window is N, the validation set for a given time t comprises samples from (t—kN+1) to t. In contrast to traditional cross-validation, NK-CV selects the last kN samples to validate the model output, avoiding random sample selection. This method is implemented to align with our emphasis on time series prediction, where the model is consistently adjusted to capture the latest characteristics of the market.

From the uppermost v% of models in each moving window, our model produces several forecasts for innovation and its direction at time t+1. Following this, we use a collective voting approach to amalgamate these forecasts and determine the projected outcomes. Each sub-model functions as an expert within a decision committee. Committee members vote for the innovation direction based on the expected innovation value. Ultimately, the address containing the most votes is chosen to be the final prediction. To enhance the accuracy of our forecasts, we integrate a threshold technique to assist the system in deciding when to generate predictions. The underlying concept is to initiate the prediction process when there are evident signs of substantial innovation (Henneke et al., 2011). Specifically, our approach will initiate the forecasting process if:

$$\frac{|positiveSign - V/2|}{V/2} > threshold \tag{8}$$

being $positiveSign$ positive number of votes produced for the models V NN.

3.4 Conceptual Overview

We examine two hypotheses. The initial hypothesis is based on the idea that the market return consists of both a linear component and a non-linear component. In cases where the market crash is computed using ARMA-GARCH, it is anticipated to exhibit non-linear characteristics (Zhang et al., 2019; Alaminos et al., 2024a). Consequently, we posit that the market yield can be represented as:

$$r_t = \mathcal{L}_t + \mathcal{N}_t \tag{9}$$

where $\mathcal{L}_t$ stands for the linear component and $\mathcal{N}_t$ for the non-linear component. Since ARMA-GARCH solely captures the linear aspects in r_t and its volatility σ_t, t an estimated bias is expected. Additionally, by breaking down $\mathcal{N}_t$ into its ARMA- and GARCH-related parts, the actual value of r_t might be understated.

$$r_t = (\widehat{c} + c_{bias}) + \sum_{i=1}^{p} (\widehat{\varphi_i} + \varphi_{bias,i})\, r_{t-1} + \varepsilon'_t + \sum_{j=1}^{q} \left(\widehat{\theta_i} + \theta_{bias,j}\right) \varepsilon'_{t-j} \quad (10a)$$

$$\varepsilon'_t = \sigma'_t z'_t, \text{ being } z'_t = z_t^{\mathcal{L}} + \mathcal{N}_t^{arma}, z_t^{\mathcal{L}} \sim N(0,1) \quad (10b)$$

$$\sigma_t'^2 = (\widehat{\omega} + \omega_{bias}) + \sum_{k=1}^{m} (\widehat{\alpha_k} + \alpha_{bias,k})\, \varepsilon_{t-k}'^2 + \sum_{l=1}^{s} \left(\widehat{\beta_l} + \beta_{bias,l}\right) \sigma_{t-l}'^2 + \mathcal{N}_t^{garch} \quad (10c)$$

being c_{bias}, $\varphi_{bias,i}$, $\theta_{bias,j}$, ω_{bias}, $\alpha_{bias,k}$, and $\beta_{bias,l}$ estimated biases of ARMA-GARCH estimators $\widehat{c}$, $\widehat{\varphi_i}$, $\widehat{\theta_i}$, $\widehat{\omega}$, $\widehat{\alpha_k}$, $\widehat{\beta_l}$ if we include a non-linear factor in the data; $\mathcal{N}_t^{arma}$ denotes the nonlinear self-dependence σ_t.

Arranging the equations mentioned earlier, we derive an expanded version of ARMA-GARCH:

$$r_t = \widehat{r}_t + \mathcal{N}_t^{arma} + \widehat{r}_{bias,t} \quad (11a)$$

$$\varepsilon'_t = \sigma'_t z'_t, \text{ being } z'_t = z_t^{\mathcal{L}} + \mathcal{N}_t^{arma}, z_t^{\mathcal{L}} \sim N(0,1) \quad (11b)$$

$$\sigma_t^2 = \widehat{\sigma}_t^2 + \mathcal{N}_t^{garch} - \widehat{\sigma}_{bias,t}^2 \quad (11c)$$

being $\widehat{r}_t$ the estimation by ARMA, and $\widehat{\sigma}$ the estimate by GARCH; $\widehat{r}_{bias,t} = c_{bias} + \sum_{i=1}^{p} \varphi_{bias,i} r_{t-1} + \sum_{j=1}^{q} \theta_{bias,j} \varepsilon_{t-j}$ symbols the estimate bias of ARMA, and $\widehat{\sigma}_{bias,t}^2 = \omega_{bias} + \sum_{k=1}^{s} \alpha_{bias,k} \varepsilon_{t-k}^2 + \sum_{l=1}^{m} \beta_{bias,l} \sigma_{t-l}^2$ incorporates the estimate bias of GARCH (Alaminos et al., 2024a).

The impact of the market shock may be assessed as (Zhang et al., 2019):

$$z_t = \frac{\mathcal{N}_t^{arma} - \widehat{r}_{bias,t}}{\sqrt{\sigma_t^2 + \mathcal{N}_t^{garch} - \widehat{\sigma}_{bias,t}^2}} \quad (12)$$

This suggests that the projected estimation of market shocks will be affected by the nonlinear self-dependency present in both ARMA and GARCH.

In principle, the ARMA-GARCH model is structured to identify the linear components of momentum in time series data. However, market upheavals are complex occurrences influenced by a multitude of factors, including macro- and micro-level economic indicators, along with the perceptions and behaviors of investors. Consequently, drawing inspiration from recent advancements in ANN and ML research, we tackle the task of predicting market disturbances through the implementing a data-driven approach.

The second proposition suggests that in the presence of non-linear patterns in the original price data, it is viable to predict the market disturbance direction like in the ARMA-GARCH model through one-step forecasting (Alaminos et al., 2024a). This involves utilizing historical price information for a sequence of market disturbances.

ARMA-GARCH methodologies might fall short in adequately capturing the non-linear components within the initial price data. Consequently, if non-linear patterns are present, they will persist in the expected market shocks. These patterns could be more effectively captured by suitable non-linear models. As a result, one-step predicting of the market shock direction becomes feasible (Sun et al., 2019). The hypotheses being tested revolve around the achieved precision rate for market shock forecasting ($\mathcal{R}$):

$$H_0 : \mathcal{R} \leq 0.5(\text{unforeseeable market impact})$$

$$H_a : \mathcal{R} > 0.5(\text{foreseeable market impact})$$

We investigate the significance of historical information in unveiling the concealed patterns that may exist beyond the scope of the ARMA-GARCH model, as outlined in our hypotheses. Several non-linear regression-based models have been proposed to address non-linear patterns in time series data.

3.5 MS-ARMA-GARCH-MLP

The input layer of the Multilayer Perceptron (MLP), a prominent category within neural networks, comprises sensory units, followed by one or more hidden layers and an output layer. When a linear input is introduced to the MLP network, it takes the form of the Hybrid MLP. The Hamilton model, viewed as a nonlinear amalgamation of autoregressive functions, which includes the multilayer perceptron, is designated as Hybrid MLP-HMC models (Olteanu et al., 2004). In the HMC model, regime changes are primarily influenced by a Markov chain, eliminating a priori assumptions regarding the number of regimes (Olteanu et al., 2004).

The Hybrid MLP fundamentally integrates network inputs, connecting them to output nodes through weighted connections, thus creating a linear model alongside the nonlinear Multilayer Perceptron. In the context of this investigation, the proposed MS-ARMA-GARCH-MLP model introduces Markov switching-type regime changes in both conditional mean and conditional variance processes. It is enhanced with MLP-type neural networks, aiming to advance in-sample and out-of-sample forecast accuracy. The outlined structure of the MS-ARMA-GARCH-MLP model is as follows (Bildirici & Ersin, 2014):

$$\boldsymbol{y_t} = \boldsymbol{c}_{(\boldsymbol{s_t})} + \sum_{\boldsymbol{i}=1}^{\boldsymbol{r}} \boldsymbol{\theta}_{\boldsymbol{i},(\boldsymbol{s_t})} \boldsymbol{y}_{\boldsymbol{t}-1} + \boldsymbol{\varepsilon}_{\boldsymbol{t},(\boldsymbol{s_t})} + \sum_{\boldsymbol{j}=1}^{\boldsymbol{n}} \boldsymbol{\varphi}_{\boldsymbol{j},(\boldsymbol{s_t})} \boldsymbol{\varepsilon}_{\boldsymbol{t}-\boldsymbol{j},(\boldsymbol{s_t})} \tag{13}$$

$$\begin{aligned} \boldsymbol{\sigma}^2{}_{\boldsymbol{t},(\boldsymbol{s_t})} &= \boldsymbol{\omega}_{(\boldsymbol{s_t})} + sum_{\boldsymbol{p}=1}^{\boldsymbol{p}} \boldsymbol{\alpha}_{\boldsymbol{p},(\boldsymbol{s_t})} \boldsymbol{\varepsilon}^2{}_{\boldsymbol{t}-\boldsymbol{p},(\boldsymbol{s_t})} + \sum_{\boldsymbol{q}=1}^{\boldsymbol{q}} \boldsymbol{\beta}_{\boldsymbol{q},(\boldsymbol{s_t})} \boldsymbol{\sigma}_{\boldsymbol{t}-\boldsymbol{q},(\boldsymbol{s_t})} \\ &+ sum_{\boldsymbol{h}=1}^{\boldsymbol{h}} \boldsymbol{\xi}_{\boldsymbol{h},(\boldsymbol{s_t})} \boldsymbol{\psi} \left(\boldsymbol{\tau}_{\boldsymbol{h},(\boldsymbol{s_t})}, \boldsymbol{Z}_{\boldsymbol{t},(\boldsymbol{s_t})} \boldsymbol{\lambda}_{\boldsymbol{h},(\boldsymbol{s_t})}, \boldsymbol{\theta}_{\boldsymbol{h},(\boldsymbol{s_t})} \right) \end{aligned} \tag{14}$$

where the states of the system are determined by an unobservable Markov process

$$\sum_{i=1}^{m} \boldsymbol{\sigma}_{t(i)}^{2} \boldsymbol{P}\left(\boldsymbol{S_t} = \boldsymbol{i} \,|\boldsymbol{z}_{t-1}\right), \boldsymbol{i} = 1, \ldots \boldsymbol{m} \tag{15}$$

In the neural network of the Multilayer Perceptron (MLP) type, the sigmoid function of logistic type is defined as:

$$\psi\left(\tau_{h(s_t)}, \boldsymbol{Z}_{t(s_t)} \boldsymbol{\lambda}_{h(s_t)}, \boldsymbol{\theta}_{h(s_t)}\right) = \left[1 + \exp\left(-\boldsymbol{\tau}_{h(s_t)}\left(\sum_{l=1}^{l}\left[\sum_{h=1}^{h} \boldsymbol{\lambda}_{h,l,(s_t)} \boldsymbol{z}_{t-l,(s_t)}^{h} + \boldsymbol{\theta}_{h(s_t)}\right]\right)\right)\right]^{-1} \tag{16}$$

$$\left(\frac{1}{2}\right) \boldsymbol{\lambda}_{h,d} \sim \boldsymbol{uniform}\left[-1, +1\right] \tag{17}$$

and $\boldsymbol{P}\left(\boldsymbol{S_t} = \boldsymbol{i} \,|\boldsymbol{z}_{t-1}\right)$, the conditional probability filtered using the following equation,

$$\boldsymbol{P}\left(\boldsymbol{S_t} = \boldsymbol{i} \,|\boldsymbol{z}_{t-1}\right) \boldsymbol{\alpha f}\left(\boldsymbol{P}\left(\boldsymbol{\sigma}_{t-1} \,|\boldsymbol{z}_{t-1},\, \boldsymbol{s}_{t-1} = 1\right)\right) \tag{18}$$

if the transition probability $\boldsymbol{P}\left(\boldsymbol{s_t} = \boldsymbol{i} \,|\boldsymbol{s}_{t-1} = \boldsymbol{j}\right)$ is acknowledged for $\boldsymbol{n}_{\boldsymbol{j},\boldsymbol{i}}$.

$$\boldsymbol{z}_{\boldsymbol{t-d}} = \frac{\left[\boldsymbol{\varepsilon}_{\boldsymbol{t-d}} - \boldsymbol{E}\left(\boldsymbol{\varepsilon}\right)\right]}{\sqrt{\boldsymbol{E}\left(\boldsymbol{\varepsilon}^{2}\right)}} \tag{19}$$

The $s \rightarrow \max\{p, q\}$ recursive procedure commences by constructing $\boldsymbol{P}\left(\mathbf{z_s} = \boldsymbol{i} \,|\boldsymbol{z}_{\boldsymbol{s}-1}\right)$, where $\psi\left(\boldsymbol{z_t}, \boldsymbol{\lambda_h}\right)$ takes the form $1/(1+\exp(-x))$. This function is twice-differentiable and continuous, with values bounded between [0, 1]. The weight vector is denoted as $\xi = w$, and $\psi = g$ represents the logistic activation function, with input variables defined as $\boldsymbol{z_t}, \boldsymbol{\lambda_h} = \boldsymbol{x_i}$, where $\boldsymbol{\lambda_h}$ is specified as in (17).

If the transition probability $\boldsymbol{P}\left(\boldsymbol{z_t} = \boldsymbol{i} \,|\boldsymbol{z}_{t-1} = \boldsymbol{j}\right)$ is accepted for $\boldsymbol{n}_{\boldsymbol{j},\boldsymbol{i}}$, then

$$f\left(y_t \,|x_t,\ z_t = i\right) = \frac{1}{\sqrt{2\pi h_{t(i)}}} exp\left\{\frac{-\left(y_t - x'_t \boldsymbol{\varphi} - \sum_{j=1}^{H} \beta_j p\left(x'_t \gamma_j\right)\right)^{2}}{2h_{t(j)}}\right\} \tag{20}$$

The recursive procedure begins with $s \rightarrow \max\{p, q\}$, by constructing $\boldsymbol{P}\left(\mathbf{z_s} = \boldsymbol{i} \,|\boldsymbol{z}_{\boldsymbol{s}-1}\right)$.

3.6 MS-ARMA-GARCH-RBF

The model MS-GARCH-RBF is specified according to Liu and Zhang (2010).

$$\boldsymbol{\sigma}_{\boldsymbol{t},(\boldsymbol{s_t})}^{2} = \boldsymbol{\omega}_{(\boldsymbol{s_t})} + \sum_{\boldsymbol{p}=1}^{\boldsymbol{p}} \boldsymbol{\alpha}_{\boldsymbol{p},(\boldsymbol{s_t})} \boldsymbol{\varepsilon}_{\boldsymbol{t-p},(\boldsymbol{s_t})}^{2} + \sum_{\boldsymbol{q}=1}^{\boldsymbol{q}} \boldsymbol{\beta}_{\boldsymbol{q},(\boldsymbol{s_t})} \boldsymbol{\sigma}_{\boldsymbol{t-q},(\boldsymbol{s_t})} + \sum_{\boldsymbol{h}=1}^{\boldsymbol{h}} \boldsymbol{\xi}_{\boldsymbol{h},(\boldsymbol{s_t})} \boldsymbol{\phi}_{\boldsymbol{h},(\boldsymbol{s_t})} \left(\left\|\boldsymbol{Z}_{\boldsymbol{t},(\boldsymbol{s_t})} - \boldsymbol{\mu}_{\boldsymbol{h},(\boldsymbol{s_t})}\right\|\right) \tag{21}$$

where $i = 1,\ldots, m$ states are determined by an unobservable Markov process:

$$\sum_{i=1}^{m} \boldsymbol{\sigma}_{t,(s_t)}^2 \boldsymbol{P}\left(\mathbf{S_t} = \boldsymbol{i} \,|\boldsymbol{z}_{t-1}\right) \tag{22}$$

A Gaussian basis function is assigned to the hidden units, denoted as (x) for $x = 1, 2,..., X$, with the activation function defined as per (Santos et al., 2010):

$$\boldsymbol{\phi}\left(\boldsymbol{h},\ (\boldsymbol{s_t})\,, \boldsymbol{Z_t}\right) = \mathbf{exp}\left(\frac{-\left\|\boldsymbol{Z}_{t,(s_t)} - \boldsymbol{\mu}_{h,(s_t)}\right\|^2}{2\boldsymbol{\rho}^2}\right) \tag{23}$$

With p defining the width of each function, $\boldsymbol{Z_t}$ represents a vector of lagged explanatory variables. The condition $\alpha+\beta<1$ is essential to ensure. Networks of this kind have the capability to generate outputs across the entire real-valued spectrum. Nevertheless, in practical scenarios where there exists prior knowledge about the expected output range, it proves more computationally efficient to apply a nonlinear transfer function to the outputs, thus incorporating that specific knowledge.

$\boldsymbol{P}\left(\mathbf{S_t} = \boldsymbol{i} \,|\boldsymbol{z}_{t-1}\right)$ denotes the filtered probability with the following representation:

$$\left(\boldsymbol{P}\left(\mathbf{S_t} = \boldsymbol{i} \,|\boldsymbol{z}_{t-1}\right) \boldsymbol{\alpha f}\left(\boldsymbol{P}\left(\boldsymbol{\sigma}_{\mathbf{t}-1} \,|\boldsymbol{z}_{t-1}, \boldsymbol{s}_{t-1} = 1\right)\right)\right) \tag{24}$$

If the transition probability $\boldsymbol{P}\left(\boldsymbol{s_t} = \boldsymbol{i} \,|\boldsymbol{s}_{t-1} = \boldsymbol{j}\right)$ for $\boldsymbol{n}_{j,i}$ is approved,

$$\boldsymbol{z}_{t-d} = \frac{\left[\boldsymbol{\varepsilon}_{t-d} - \boldsymbol{E}(\boldsymbol{\varepsilon})\right]}{\sqrt{\boldsymbol{E}(\boldsymbol{\varepsilon}^2)}} \tag{25}$$

The recursive procedure begins with $s \rightarrow \max\{p, q\}$, and it commences by constructing $\boldsymbol{P}\left(\mathbf{z_s} = \boldsymbol{i} \,|\boldsymbol{z}_{s-1}\right)$.

3.7 Fractal Properties into the NN–ARMA–GARCH Model

(1) The model begins with the innovations z_{tz} from the ARMA(1,1)–GARCH(1,1) specification, which contain the unexplained portion of market behavior. These innovations are treated as signals with nonlinear and multiscale structure.

(2) Once the ARMA(1,1)–GARCH(1,1) model is estimated, the standardized innovation series z_{tz} is obtained, capturing the nonlinear and volatile components not explained by the conditional mean and variance. Rather than applying classical fractal analysis methods (such as estimating fractal dimension), the manuscript's approach implements an operational fractal simulation through the hierarchical organization of variables derived from z_{tz}. The procedure is as follows:

- Lagged series of z_{tz}, σ_t (conditional volatility), and other derived indicators (such as signs, returns, realized variance, bi-power variance, true range, etc.) are generated with lags ranging from 1 to 60.

- These variables are empirically grouped into lag segments representing different time scales:

  Scale 1 (lags 1–10): market microstructure (very short-term),
  Scale 2 (lags 11–30): intermediate structure,
  Scale 3 (lags 31–60): persistent or structural effects.

- Each scale is assigned a distinct block of variables, which are not mixed or combined with others. Each block is treated as an independent input into the neural network.
- These scales function as simulated fractal layers, preserving the informational independence of each level, allowing the network to learn hierarchical and self-similar patterns, akin to a system observing a natural fractal structure.

(3) Each scale is processed separately within architectures specifically designed to handle fractal structures:

- F-DRCNN combines scales through convolutions and recurrence, weighted by their relevance λ_n.
- **F-DNDT m**akes smoothed decisions across scales using soft binning.
- **F-QNN and F-QRNN e**ncode each scale into quantum rotation angles and combine hierarchical states through recurrent mechanisms.

(4) A multiscale loss function is applied, where each scale is assigned a weight λ_n based on its informational importance, enabling the model to learn patterns at different levels of temporal complexity.

3.8 ARMA-GARCH-Fractal Deep Recurrent Convolutional Neural Network (ARMA-GARCH-F-DRCNN)

After the prior advancement of ARMA-GARCH, RNNs have been applied in various domains for successful time series forecasting, thanks to their significant predictive capability. The conventional RNN framework is organised based on its output, which is influenced by its preceding estimates (Wan et al., 2017). Formulas (26) and (27) can be utilised to derive an input sequence vector x, the hidden states of a recurrent layer s, and the output of a singular hidden layer y.

$$s_t = \sigma\left(W_{xs}x_t + W_{ss}s_{t-1} + b_s\right) \tag{26}$$

$$y_t = o\left(W_{so}s_t + b_y\right) \tag{27}$$

Denoting the weights 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 W_{xs}, W_{ss}, and W_{so} respectively. The biases of both the hidden layer and the output layer represented by b_y in Formula (28), symbols σ and o indicate the activation functions (Alaminos et al., 2024b).

$$STFT\left\{z(t)\right\}(\tau,\omega)=\int_{-\infty}^{+\infty}z(t)\omega\left(t-\tau\right)e^{-j\omega t}dt\Big) \tag{28}$$

Vibration signals are represented by *z(t),* while symbols *ω(t)* signify the incorporation of innovations from the ARMA-GARCH. A complex function, denoted as *T(τ, ω),* defines the vibration signals across time and frequency. The computation of hidden layers involves the use of the next equations.

$$S_t=\sigma\left(W_{TS}*T_t+W_{SS}*S_{t-1}+B_s\right) \tag{29}$$

$$Y_t=o\left(W_{YS}*S_t+B_y\right) \tag{30}$$

where W is the convolution kernels.

To create a deep structure, the recurrent convolutional neural network (RCNN) can be arranged in layers, resulting in the formation of the DRCNN (Alaminos et al., 2024c; Huang & Narayanan, 2017). In this composite scenario, the final segment of the model constitutes a supervised learning layer, as defined by formula (31).

$$\widehat{r}=\sigma\left(W_h*h+b_h\right) \tag{31}$$

with W_h representing the weight and b_h representing the bias, respectively, the disparity between predicted and actual observations in the training data for prediction can be assessed. This difference may then be used to provide feedback for model training (Mancini et al., 2022). The distribution of the output of the ARMA-GARCH model is denoted as $\widehat{\boldsymbol{r}}_{\text{ARMA–GARCH}}$. The formulation of the integration is defined as follows, according to García García et al. (2012):

$$\varepsilon\left(t\right)=\sigma(t)\bullet\omega\left(t\right) \tag{32}$$

$$\sigma^2\left(t\right)=\sum_{k=1}^{p}\alpha_k\varepsilon^2\left(t-k\right)+\sum_{l=1}^{q}\beta_l\sigma^2(t-l) \tag{33}$$

$$z\left(t\right)=\sum_{i=1}^{p}\phi_i z\left(t-i\right)+\sum_{j=1}^{q}\theta_j\sigma^2\left(t-j\right)+\varepsilon_t \tag{34}$$

The error term ε(t) at time t reflects the discrepancies between predictions and observed values. The conditional standard deviation σ(t) and the innovation ω(t) capture variability and unpredictable information in the model, respectively. The parameters α_k and β_l of the GARCH model dictate the influence of past errors and variations on conditional variance. Additionally, the coefficients ϕ_i and θ_j of the AutoRegressive (AR) and Moving Average (MA) models represent the influences of past values and terms in the time series z(t).

The Fractal Deep Recurrent Convolutional Neural Network (F-DRCNN) integrates fractal dynamics into deep learning by combining multiscale convolutional operations with recurrent structures. This architecture enables the model to capture both spatial and temporal dependencies across multiple hierarchical scales, making it highly effective for complex, multiscale data such as time series, and spatiotemporal

datasets (Roberto et al., 2021). Fractal convolutional layers operate at different scales n, capturing spatial features hierarchically. The operation is defined as:

$$\boldsymbol{s}_{\boldsymbol{t}}^{(\boldsymbol{n})} = \boldsymbol{\sigma}(\boldsymbol{W}_{\boldsymbol{c}}^{(\boldsymbol{n})} * \boldsymbol{s}_{\boldsymbol{t}-1}^{(\boldsymbol{n})} + \boldsymbol{b}_{\boldsymbol{c}}^{(\boldsymbol{n})}) \tag{35}$$

where the feature map $\boldsymbol{s}_{\boldsymbol{t}}^{(\boldsymbol{n})}$ represents the transformed input at time t and scale n, $\boldsymbol{W}_{\boldsymbol{c}}^{(\boldsymbol{n})}$ is the convolution kernel and $\boldsymbol{b}_{\boldsymbol{c}}^{(\boldsymbol{n})}$ is the bias term at scale n. The activation function σ, using tanh, introduces nonlinearity to enhance feature representation (Raubitzek & Neubauer, 2021).

Feature aggregation combines information from multiple scales to produce the final representation r_t. This is achieved by summing the weighted contributions from all scales:

$$\boldsymbol{r}_{\boldsymbol{t}} = \sum_{\boldsymbol{n}=1}^{\boldsymbol{N}} \boldsymbol{\lambda}_{\boldsymbol{n}} \bullet \boldsymbol{Concat}(\boldsymbol{s}_{\boldsymbol{t}}^{(\boldsymbol{n})}, \boldsymbol{h}_{\boldsymbol{t}}^{(\boldsymbol{n})}) \tag{36}$$

where $\boldsymbol{s}_{\boldsymbol{t}}^{(\boldsymbol{n})}$ represents the feature map at time t and scale n, $\boldsymbol{h}_{\boldsymbol{t}}^{(\boldsymbol{n})}$ captures additional hierarchical or auxiliary features at the same scale, and λ_n is the weighting coefficient that controls the contribution of scale n (Li & Zhou, 2022). The concatenation operation $\boldsymbol{Concat}(\boldsymbol{s}_{\boldsymbol{t}}^{(\boldsymbol{n})}, \boldsymbol{h}_{\boldsymbol{t}}^{(\boldsymbol{n})})$ merges the feature maps and hierarchical representations. The fractal loss function optimizes predictions across all scales, ensuring multiscale learning:

$$\boldsymbol{L} = \sum_{\boldsymbol{n}=1}^{\boldsymbol{N}} \boldsymbol{\lambda}_{\boldsymbol{n}} \bullet \boldsymbol{L}_{\boldsymbol{n}} \tag{37}$$

where a scale-specific loss L_n is computed for each fractal scale n, reflecting the prediction error at that scale (Roberto et al., 2021). The overall contribution of each scale to the optimization process is controlled by the weighting coefficient λn, which balances the importance of scales:

$$\boldsymbol{\lambda}_{\boldsymbol{n}} = \frac{1}{\boldsymbol{Z}} \bullet \boldsymbol{e}^{-\boldsymbol{\beta} \bullet \boldsymbol{Var}(\boldsymbol{L}_{\boldsymbol{n}})} \tag{38}$$

being $\boldsymbol{Var}(\boldsymbol{L}_{\boldsymbol{n}})$, the variance of the loss L_n at scale n, which quantifies the significance of that scale in contributing to the overall model's learning process. The factor β is a hyperparameter that adjusts the penalty for low-variance scales (Roberto et al., 2021). To ensure proper weight distribution across scales, the coefficients λ_n are normalized using a constant Z, enforcing the constraint $\sum_{\boldsymbol{n}=1}^{\boldsymbol{N}} \boldsymbol{\lambda}_{\boldsymbol{n}} = 1$.

3.9 ARMA-GARCH-Fractal Deep Neural Decision Trees (ARMA-GARCH-F-DNDT)

Decision Tree models, abbreviated as DNDTs, are integrated into deep-learning neural networks. They utilize stochastic gradient descent (SGD) for optimizing all parameters, steering clear of a complex and greedy partitioning process. This approach facilitates efficient large-scale processing through mini-batch-based learning, seamlessly integrating with any broader neural network (NN) model. This allows for end-to-end learning with backpropagation. In contrast to traditional Decision Trees (DTs) that utilize a greedy, recursive feature partitioning method (Alaminos et al., 2019; Quinlan, 1993), DNDTs utilize a soft binning function to evaluate error rates for each node. This approach streamlines decision-making, resulting in the formation of DNDTs (Dougherty et al., 1995). Typically, the input to the binning function is a real scalar'x,'creating an index for the containers to which'x'belongs (Alaminos et al., 2024b). In the case of'x'being a continuous variable, it is grouped into'n + 1'intervals, necessitating'n'trainable cut-off points in this context. The cut-off points are denoted as $(\beta_1, \beta_2, \ldots, \beta_n)$ and are strictly ascending such that $\beta_1 < \beta_2 < \ldots < \beta_n$.

The DNDT algorithm activation function relies on the NN outlined in next equation according to Dougherty et al. (1995):

$$\pi = fw, \omega\,(t)\,, b, \tau\,(x) = softmax((wx\omega + b)/\tau) \tag{39}$$

where w ilustrates $w = [1, 2, \ldots, n+1]$, $\tau > 0$ represents a temperature factor, and b is schown in the next Equation.

$$b = [0, -\beta_1, -\beta_1, -\beta_2 \ldots, -\beta_1 - \beta_2 - \cdots - \beta_{n]} \tag{40}$$

if τ tends to 0, the vector sampling is incorporataed by the Straight-Through (ST) Gumbel–Softmax approach (Alaminos et al., 2024b).

Following the previously mentioned binning function, the fundamental concept involves constructing the DT by the Kronecker product. Let's consider an input instance $x \in R^D$ having D features. By relating every feacture x_d with its NN $f_d(x_d)$, all terminal nodes of the DT can be identified using the next formula, as outlined in (Ho, 1998):

$$z = f_1\,(x_1) \otimes f_2\,(x_2) \otimes \cdots \otimes f_D(x_D) \tag{41}$$

here z denotes a vector indicating the index of the leaf node reached by the instance x. The complexity parameter of the model, determining the number of cut points per feature, is unrestricted. For instance, these cut points could be smaller than the minimum value of x_d or larger than its maximum value (Alaminos et al., 2024b).

The ARMA-GARCH process output as y_t and the output of the Decision Tree as z. The joint distribution can be defined as $\boldsymbol{P}\left(\boldsymbol{y_t}, \boldsymbol{z}\right) = \boldsymbol{P}\left(\boldsymbol{y_t}\,|\boldsymbol{z}\right).\boldsymbol{P}\left(\boldsymbol{z}\right)$. The ARMA-GARCH process distribution, indicated as $\boldsymbol{P}\left(\boldsymbol{y_t}, \boldsymbol{z}\right)$, is commonly assumed to follow a Gaussian distribution. Considering the historical data of the time series up to time

t-1 and the information derived from the Decision Tree z, the distribution of $\boldsymbol{y_t}$ can be expressed according to Ho (1998) as:

$$\boldsymbol{P}\left(\boldsymbol{y_t}\,|\boldsymbol{z}\right) = \boldsymbol{N}\left(\boldsymbol{\mu_t}, \boldsymbol{\sigma_t^2}\right) \tag{42}$$

where N signifies the normal distribution, and $\boldsymbol{\mu_t}$ and $\boldsymbol{\sigma_t^2}$ represent the mean and variance parameters, respectively. These parameters are influenced by previous observations and GARCH parameters. The specific expressions for $\boldsymbol{\mu_t}$ and $\boldsymbol{\sigma_t^2}$ rely on the particular ARMA-GARCH model and its associated parameters.

The distribution of the output from the Decision Tree, indicated as P(z), is intricately influenced by the synergy between the soft binning function and the structure involving the Kronecker product. The specific features of this distribution depend on the complexities of the binning function and the fundamental design of the Decision Tree. For example, when z represents a categorical variable indicating the leaf node index, P(z) takes on the form of a categorical distribution (Alaminos, Salas & Partal-Ureña, 2024a, 2024b, 2024c, 2024d, 2024e; Ho, 1998):

$$P\left(z\right) = Categorical\left(\pi_1, \pi_2, \ldots, \pi_k\right) \tag{43}$$

Being π_i the probability of reaching the i-th leaf node is determined by the interplay of the soft binning function and the structure involving the Kronecker product.

The Fractal Neural Decision Tree (F-DNDT) is an enhancement of traditional Decision Neural Trees (DNDT) that incorporates fractal dynamics to enable multi-scale decision-making (Li & Zhou, 2022). By introducing fractal scaling, the model captures both fine-grained and coarse-grained feature interactions across hierarchical structures.

$$\widehat{\boldsymbol{y}}_{\boldsymbol{t}} = \sum_{\boldsymbol{n}=1}^{\boldsymbol{N}} \boldsymbol{\lambda_n} \prod_{\boldsymbol{d}=1}^{\boldsymbol{D}} \boldsymbol{\phi_d^{(n)}}(\boldsymbol{x_t}) \tag{44}$$

where predicted output $\widehat{\boldsymbol{y}}_{\boldsymbol{t}}$ represents the model's estimation for the input x_t at time t. The model leverages a total of N fractal scales and D features (or dimensions) to capture intricate patterns in the data. For the d-th feature at scale n, the fractal-based soft binning function, $\varnothing_{\boldsymbol{d}}^{(\boldsymbol{n})}(\boldsymbol{x_t})$, encodes input information into a fractal representation (Alaminos et al., 2024c; Li & Zhou, 2022). Each fractal scale n is associated with a weighting coefficient λ_n, which governs the relative contribution of that scale to the final prediction (Raubitzek & Neubauer, 2021).

Traditional DNDTs rely on hard thresholds or simple linear splits to partition the feature space. F-DNDT replaces these with fractal-based soft binning functions, enabling smoother and multiscale decision boundaries:

$$\varnothing_{\boldsymbol{d}}^{(\boldsymbol{n})}\left(\boldsymbol{x_t}\right) = \boldsymbol{\sigma}(\boldsymbol{a_d^{(n)}}\boldsymbol{x_t} + \boldsymbol{b_d^{(n)}}) \tag{45}$$

being the d-th feature at scale n, the fractal scale parameter $a_d^{(n)}$ determines the granularity of the fractal representation, while the bias term $b_d^{(n)}$ adjusts the soft binning at each scale. To ensure smooth, probabilistic splits that are differentiable, the model employs activation functions such as sigmoid or softmax (σ) (Li & Zhou, 2022). By stacking multiple scales (n), the soft binning functions capture intricate interactions between features and hierarchical dependencies (Roberto et al., 2021). To adapt to fractal complexity, each scale is assigned a weighting coefficient λ_n:

$$\boldsymbol{\lambda_n} = \frac{1}{\boldsymbol{Z}} \bullet \boldsymbol{e}^{-\boldsymbol{\beta} \bullet \boldsymbol{Var}(\varnothing^{(n)})} \tag{46}$$

where the variance of the soft binning outputs at scale n, $\boldsymbol{Var}(\varnothing^{(n)})$, quantifies the importance of that scale by reflecting its contribution to feature representation. A scaling factor β introduces a penalty for scales with low variance, encouraging the model to focus on scales with higher informational content. To ensure proper weight distribution across scales, the coefficients λ_n are normalized using a constant Z, enforcing the constraint $\sum_{n=1}^{N} \boldsymbol{\lambda_n} = 1$.

The Fractal-Driven Neural Decision Tree (F-DNDT) establishes a hierarchical framework where each node integrates decisions derived from multiple fractal scales. This multi-scale approach empowers the model to seamlessly blend local and global perspectives (Raubitzek & Neubauer, 2021). At finer scales, the model captures detailed, feature-specific information, facilitating precise, leaf-level decisions. Meanwhile, coarser scales extract broader, high-level patterns that inform global decision-making (Li & Zhou, 2022). The loss function integrates the errors across all scales to ensure multiscale optimization:

$$\boldsymbol{L} = \sum_{\boldsymbol{n}=1}^{\boldsymbol{N}} \boldsymbol{\lambda_n} \bullet \boldsymbol{CrossEntropy}(\widehat{\boldsymbol{y}}_{\boldsymbol{t}}^{(\boldsymbol{n})}, \boldsymbol{y_t}) \tag{47}$$

where $\widehat{\boldsymbol{y}}_{\boldsymbol{t}}^{(\boldsymbol{n})}$ is the prediction at scale n, and y_t is the ground truth (Roberto et al., 2021).

3.10 ARMA-GARCH-Fractal Quantum Neural Network (ARMA-GARCH-F-QNN)

The combination of CNNs and quantum computing holds the potential to provide a computational strategy with significant predictive capabilities (Wan et al., 2017). In quantum computing, the smallest unit of information is a qubit, providing a probabilistic representation. A qubit can exist in a state of either"1"or"0"or any superposition of the two (Dos Santos Gonçalves, 2019). Equation (48) defines the state of a qubit:

$$|\psi\rangle = \alpha\,|o\rangle + \beta\ |1\rangle \tag{48}$$

Given that α and β are the values indicating the amplitude of their respective states, ensuring that $|\alpha|^2 + |\beta|^2 = 1$. These values are represented as a pair of numbers

$\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ The angle θ signifies the specification of geometric aspects and is identified as $\cos(\theta) = |\alpha| \, and \sin(\theta) = |\beta|$. Quantum gates is employed to set probabilities through weight enhancement (Zidan et al., 2019). An example of a rotation gate is presented in formula (49).

$$U(\Delta\theta) = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \tag{49}$$

The quantum gate mentioned earlier enables the enhancement of a qubit's state. The utilization of the spin gate on a qubit is outlined as follows, as per Zidan et al. (2019):

$$\begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{50}$$

The procedure begins with a quantum hidden neuron derived from the state $|o\rangle$, setting up the superposition as detailed in the next Equation.

$$\sqrt{p}\left|O\right\rangle + \sqrt{1-p}\,|1\rangle with O \leq |p| \leq 1 \tag{51}$$

The random probability of initiating the system in the state $|O\rangle$ is expressed by p. Random number generation induces the classical neurons (Beer et al., 2020; Zidan et al., 2019). The establishment of the outcome from the quantum neuron is detailed as per Formula (52).

$$v_j = f\left(\sum_{i=1}^{n} w_{ji} * x_i * \omega(t)\right) \tag{52}$$

f represents a problem-dependent sigmoid or Gaussian function and $\omega(t)$ is the signals of the ARMA-GARCH equation.

The subsequent equation signifies the portrayal of Quantum Neural Network (QNN) predictions, distinguished by their probabilistic essence, alongside the outputs of the ARMA-GARCH model. The combined distribution adeptly encapsulates the collective uncertainties inherent in both models. Moreover, the introduction of a weighting factor (α) facilitates dynamic adjustments, empowering practitioners to finely calibrate the impact of each model P(YIntegrated) ∝ P(YQNN)·P(YARMA-GARCH) (Alaminos et al., 2024d).

The output from the network is described in Eq. (53) according to Beer et al. (2020), Liu and Zhang (2010), Alaminos, Salas and Partal-Ureña (2024a, 2024b, 2024c, 2024d, 2024e):

$$y_k = f\left(\sum_{j=1}^{l} w_{jk} * v_j\right) \tag{53}$$

The desired output is the o_k and the upgrading of output layer weight is shown as:

$$E_k^2 = \frac{1}{2}|y_k - o_k|^2 \tag{54}$$

$$\Delta w_{jk} = \eta e_k f' v_j \tag{55}$$

The output of the Fractal QNN is a summation of scale-weighted quantum probabilities derived from quantum states (Li & Zhou, 2022):

$$\widehat{y}_t = \sum_{n=1}^{N} \lambda_n \bullet |\cos(\theta_n x_t) + i sin(\theta_n x_t)|^2 \tag{56}$$

where y_t is the predicted output for the input x_t at time t, λ_n is the weighting coefficient for fractal scale n, dynamically scaling the importance of each quantum operation, and θ_n is the quantum rotation angle at scale n, modulating feature transformations across scales. The rotation angle is fractal-weighted and adapts dynamically based on the input:

$$\theta_n = \boldsymbol{\lambda_n} \bullet AmplitudeScaling(x_t) \tag{57}$$

being $AmplitudeScaling(x_t)$ a function mapping the input x_t to quantum amplitudes, normalized to [0,2π], and $\boldsymbol{\lambda_n}$ represents the fractal scale weight, determined by the importance of scale n (Raubitzek & Neubauer, 2021).

The λ_n coefficients ensure that the quantum computations adapt to the hierarchical relevance of each scale. Scales with higher relevance contribute more significantly to the final prediction. The coefficients are normalized across scales to maintain their interpretability and ensure $\sum_{n=1}^{N} \lambda_n = 1$. Each quantum state is represented as a combination of real and imaginary components, with the predicted value derived from the squared amplitude:

$$P_n = |\cos(\theta_n x_t) + i sin(\theta_n x_t)|^2 = \cos(\theta_n x_t)^2 + sin(\theta_n x_t)^2 \tag{58}$$

The use of cos and sin ensures the preservation of quantum probability normalization ($P_n \in [0,1]$). The function AmplitudeScaling(x_t) maps the input x_t to amplitudes compatible with quantum rotations. This can be implemented as:

$$AmplitudeScaling(x_t) = \frac{x_t - \min(x)}{\max(x) - \min(x)} \bullet 2\pi \tag{59}$$

where x_t denotes the input value, and min(x) and max(x) are minimum and maximum values in the input space. The fractal loss function ensures optimization across all scales:

$$\boldsymbol{L} = \sum_{\boldsymbol{n}=1}^{\boldsymbol{N}} \boldsymbol{\lambda_n} \bullet \boldsymbol{L_n} \tag{60}$$

where L_n is the scale-specific loss, following Cross-Entropy (Roberto et al., 2021).

3.11 Quantum Fractal Recurrent Neural Network (ARMA-GARCH-F-QRNN)

Within the n-fold Hikbert space of the tensor product $\mathcal{H} = \left(\mathbb{C}^2\right)^{\otimes d}$, a quantum system consisting of n qubits exists, resulting in a dimension of 2^d. A quantum state, typically despicted by a unit vector $\psi \in \mathcal{H}$, is expressed in quantum computing using bra-ket notation as $|\psi\rangle \in \mathcal{H}$. Its conjugate transpose is denoted by $\langle \psi | = | \psi \rangle^{\dagger}$, and the inner product $\langle \psi | \psi \rangle = \|\psi\|_2^2$ represents the square of the 2-norm of ψ. The outer products is designated by $[\psi\rangle\langle\psi]$, resulting in a tensor of rank 2. Foundational computational states, such as to $|0\rangle = (1, 0)$, $|1\rangle = (0, 1)$, as well as composite ground states like $|01\rangle = |0\rangle \otimes |1\rangle = (0, 1, 0, 0)$, are defined (Alaminos et al., 2024d; Beer et al., 2020; Mahajan, 2011).

A quantum gate can be understood as a unitary operation, denoted by $\mathbb{U}$ acting on a Hilbert space$\mathcal{H}$. This operation operates specifically on a subsetof qubits $\mathbb{S}$, represented as $\mathbb{S} \subseteq$ [n]. The unitary operation$\mathbb{U}$, is a nontrivial operation on this subset, denoted as $\mathbb{U} \epsilon \mathbb{S}\mathbb{U}\left(2^{|\mathbb{S}|}\right)$. to To apply $\mathbb{U}$ to the entire Hilbert space $\mathcal{H}$, it is to act as the identity on the remaining qubits, expressed as $\mathbb{U}_{\mathbb{S}} 1_{[n]\setminus\mathbb{S}}$. This extension is often overlooked, and its relevance is in indicating whether the gate is part of a quantum circuit. For instance, the initial gate R(θ) in a circuit signifies a unitary operation on a qubit, affecting the second qubit below it. This operation depends on the parameter θ. The dotted line extending from the gate implies a"controlled"operation. If the control affects only a single qubit, it denotes a block-diagonal unitary map given by$|0\rangle\langle 0| \otimes 1 + |1\rangle\langle 1| \otimes \mathbb{R}(\theta) = 1 \oplus \mathbb{R}(\theta)$. This representation means "apply $\mathbb{R}(\theta)$ if the control qubit is in state $|1\rangle$".Gate sequences are computed as matrix products within circuits, as discussed by Bausch (2020).

The projective measurements of a single qubit are determined by a Hermitian 2×2 matrix, denoted as P, for example, M $|1\rangle\langle 1|$ = diag(0, 1). The complementary outcome is then represented as $M^{\perp} = 1 - \mathrm{M}$. These measurements are quantified in meters within the circuit (Alaminos et al., 2023). Considering a quantum state represented by $|\psi\rangle$, the post-measurement state is given by $M|\psi\rangle/p$/p with probability $p = \|M|\psi\|_2$. This probability also serves as the post-selection likelihood to ensure the measured result M. This likelihood can be increased close to 1 through approximately$\sim \sqrt{1/p}$ rounds of amplitude amplification, as discussed by Herman et al. (2022).

The quantum recurrent neural networks elucidated in this proposal are compatible with conventional computing systems. In this scenario, the"hidden state"portrayed across n qubits is illustrated as an array with a magnitude of 2^n. The parameters emanate from an assemblage of all parameterized quantum gates involved in the pro-

cedure, resulting in matrices featuring parameterized inputs. In the customary execution of a Quantum Recurrent Neural Network (QRNN), a sequence of matrix–vector multiplications is utilised for the gates, succeeded by matrix–vector multiplications and subsequent normalisation of the state to attain a norm of 1 for measurement and post-selection transactions (Biamonte et al., 2017; Herman et al., 2022).

We construct a highly organized parameterized quantum circuit, where a limited number of parameters are repetitively employed, as outlined by Herman et al. (2022). This circuit is predominantly centered around a novel kind of quantum neuron that adjusts its target trajectory based on a non-linear activation function linked to the polynomials of its inputs. The configuration encompasses an initial phase that, at each iteration, assimilates the present input into the cell's state. The iterative deployment of these Quantum Recurrent Neural Network (QRNN) cells across a sequence of inputs in a recurrent model closely parallels the methodology employed in conventional Recurrent Neural Networks (RNNs). During the training process, we employ quantum amplitude amplification, as described by Biamonte et al. (2017), on the output paths.

However, nonlinear behaviour is not observed in quantum mechanics. A straightforward example is a single-qubit gate R(θ)=exp(iYθ) involving the Pauli matrix Y (Benedetti et al., 2019; Grover, 2005; Sun et al., 2019), serving as an illustration:

$$R(\theta) = \exp\left(i\theta \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\right) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \tag{61}$$

specifically, envisioning a rotation within the two-dimensional space spanned by the computational basis vectors of a single qubit, $\{|0\rangle, |1\rangle\}$, is akin to a rotation. Despite the linearity of the rotation matrix itself, it's noteworthy that the state amplitudes — $\cos\theta$ and $\sin\theta$— exhibit a non-linear dependence on the angle θ. Extending this rotation to a controlled operation, denoted as cR(i,θ_i) contingent on the i^{th} qubit of a state $|x\rangle$ for$x \in \{0,1\}^n$, we obtain the next map (Endo et al., 2021; Sun et al., 2019):

$$R(\theta_0)\, cR(1,\theta_1) \ldots cR(n,\theta_n)\, |x\rangle|0\rangle = |x\rangle\, (cos\,(\eta)\,|0\rangle + sin\,(\eta)\,|1\rangle)\ \text{for}\eta = \theta_0 + \sum_{i=1}^{n} \theta_i x_i \tag{62}$$

Therefore, this equates to a rotation through an infinite transformation of the basis vector $|x\rangle$ with$x = \{x_{1,\ldots,}x_n\} \in \{0,1\}^n$, governed by a parameter vector $\theta = (\theta_0, \theta_1, \ldots, \theta_n)$. The systematic extension of the procedure applies to both the superpositions of the base and target states. Owing to the nature of R(θ), all introduced alterations in amplitude are indeed real-valued.

The introduction of non-linearity occurs inherently through a controlled operation in transforming cosine amplitudes; however, the sine function lacks a pronounced sharpness and does not possess a sufficiently"flat"region where the activation remains constant, similar to a linear rectified unit. Guerreschi (2019) proposes a method to achieve a linear mapping onto a set of qubits, resulting in amplitudes characterized by steeper inclines and plateaus, reminiscent of a sigmoidal activation function. The activation includes a parameter, denoted as order (ord), with a value greater than or equal to 1, determining the inclination of the circuit and thus influencing the activa-

tion amplitude. In this quantum neuron, operating on pure states, a rotation angle $f(\theta) = \arctan\left(tan(\theta)^{2ord}\right)$, where ord (order) exceeds or equals 1. For the input x_i bitstring x_i, assuming an affine transformation η as articulated in formula (63), rotation is later converted via the amplitudes.

$$cos(f(\eta)) = \frac{1}{\sqrt{1 + tan(\eta)^{2x2^{ord}}}} \text{and } sin(f(\eta)) = \frac{tan(\eta)^{2^{ord}}}{\sqrt{1 + tan(\eta)^{2x2^{ord}}}} \quad (63)$$

which comes by standardising the transform $|0\rangle \mapsto\!\!\!\rightarrow cos(\theta)^{2^{ord}}|0\rangle + sin(\theta)^{2^{ord}}|1\rangle$ as can be seen clearly. For ord = 1, the circuit is displayed on the left, while on the right is the circuit for ord = 2. Higher orders can be recursively constructed. Starting with a pure state (e.g. $|x\rangle$ for $x \in \{0,1\}^2$ and repeating the process every time a measurement yields 1, an arbitrarily high probability of success is achieved.

For managing control in superposition scenarios, like $|x\rangle + |y\rangle/\sqrt{2}$, where $x \neq y$ and both are two bit-strings of length n, a direct application is not efficient. The amplitudes within the overlap in this situation are contingent upon the success outcome. To tackle this, a methodology termed fixed-point oblique amplitude amplification (Endo et al., 2021) is utilised. The Fractal Quantum Recurrent Neural Network (F-QRNN) enhances the QRNN framework by introducing fractal dynamics into the modeling process. In the fractal extension, the Hilbert space $H = (C^2)^{\otimes d}$ incorporates fractal scaling across different scales n:

$$H^{(n)} = (C^2)^{\otimes d} \quad (64)$$

where d_n varies for each scale n to capture hierarchical quantum state representations (Raubitzek & Neubauer, 2021). The total state space aggregates states across scales:

$$|\psi\rangle = \sum_{n=1}^{N} \lambda_n \left|\psi^{(n)}\right\rangle \quad (65)$$

being $\left|\psi^{(n)}\right\rangle \in H^{(n)}$ and λ_n are fractal weighting coefficients (Roberto et al., 2021). Fractal dynamics are embedded into the rotation angle θ_n of quantum gates:

$$\theta_n = \lambda_n \bullet AmplitudeScaling(x_t) \quad (66)$$

where $\boldsymbol{\lambda_n} = \frac{1}{\boldsymbol{Z}} \bullet \boldsymbol{e}^{-\boldsymbol{\beta} \bullet \boldsymbol{Var}(\boldsymbol{L_n})}$ dynamically scales the importance of each fractal scale n, based on the variance of the scale-specific loss L_n, and AmplitudeScaling(x_t) maps classical inputs to quantum amplitudes (Li & Zhou, 2022). The parameterized fractal rotation gate becomes:

$$\boldsymbol{R}^{(n)}(\boldsymbol{\theta_n}) = \begin{bmatrix} \boldsymbol{cos}(\boldsymbol{\theta_n}) & \boldsymbol{isin}(\boldsymbol{\theta_n}) \\ \boldsymbol{isin}(\boldsymbol{\theta_n}) & \boldsymbol{cos}(\boldsymbol{\theta_n}) \end{bmatrix} \quad (67)$$

The hidden state of the F-QRNN is a component that integrates fractal scaling to model hierarchical and multiscale dependencies effectively. Unlike traditional recurrent neural networks, where the hidden state is a single-dimensional representation of past inputs, the F-QRNN expands this concept by incorporating contributions from multiple hierarchical scales, leveraging fractal principles (Li & Zhou, 2022; Raubitzek & Neubauer, 2021). The hidden state is defined as:

$$h_t = \sum_{n=1}^{N} \lambda_n h_t^{(n)} \tag{68}$$

where $h_t^{(n)}$ is the hidden state at scale n, computed using:

$$h_t^{(n)} = \sigma(W_h^{(n)} h_{t-1}^{(n)} + U_h^{(n)} s_t^{(n)} + b_h^{(n)}) \tag{69}$$

being $s_t^{(n)}$ is the scale-specific spatial feature from the ARMA-GARCH model, σ denotes nonlinear quantum activation function, such as $\mathrm{f}(\eta) = \arctan\left(\tan(\eta)^{2ord}\right)$, where *ord* determines the steepness of the activation.

We enhance the capabilities of the quantum neuron by introducing additional check terms. Specifically, the term η, as outlined in formula (52), is an affine transformation of the boolean vector $x = \{x_{1,\ldots,}x_n\}$ for $x_i \in \{0, 1\}$. By incorporating multi-control gates that have their parameterized rotations—incorporating signals from ARMA-GARCH equations—designated by a multi-index θ_I, varying based on the qubits $i \in I$ where the gate conditions apply, we open the possibility of integrating higher-degree polynomials (Alaminos, Salas & Partal-Ureña, 2024a, 2024b, 2024c, 2024d, 2024e; Nielsen & Chuang, 2001; Sarker, 2021).

$$\eta' = \theta_0 + \sum_{i=1}^{n} \theta_i x_i \omega^{(n)}(t) + \sum_{i=1}^{n}\sum_{j=1}^{n} \theta_{ij} x_i x_j \omega^{(n)}(t) + \cdots = \sum_{\substack{I \subseteq [n] \\ |I| \leq d}} \theta_I \prod_{i \in I} x_i \omega^{(n)}(t) \tag{70}$$

For a neuron with a degree represented by'd', consider an illustration where d=2 and n=4. In this scenario, an examined rotation serves as an instance of elevating to a higher-order transformation denoted as η' on the bit string x_i. Consequently, more intricate boolean logic operations of higher degrees can be encapsulated directly within a singular conditional rotation. For instance, an AND operation between two bits x_1 and x_2 is effortlessly expressed as the product $x_1 x_2$.

The ARMA-GARCH process is defined by the equation $y_t = \mu_t + \epsilon_t y_t$, where the error term is determined by $\epsilon_t = \sigma_t z_t \epsilon_t$. The conditional variance is intricately modeled through the equation $\sigma_t^2 = \omega + \alpha\epsilon_t^2 - 1 + \beta\sigma_t^2 - 1$. which incorporates past information. Ultimately, the probabilistic description of the process is articulated as $P\left(y_t \,|pastinformation\right) = N\left(\mu_t \sigma_t^2\right)$, conforming to a normal distribution with mean μ_t and variance σ_t^2. The probability density function (fARMA-GARCH) for an

ARMA-GARCH process with a Gaussian distribution is expressed as per Ling and McAleer (2003), Tacchino et al. (2019):

$$f_{ARMA-GARCH}\left(y_t;\mu_t,\sigma_t^2\right)=\frac{1}{\sqrt{2\pi\sigma_t^2}}\exp(-\frac{\left(y_t-\mu_t\right)^2}{2\sigma_t^2})\bullet\frac{1}{\exp(-2\sigma_t^2\left(y_t-\mu_t\right)^2)} \quad (71)$$

To put into action the created QRNN cell, as per Barndorff-Nielsen and Shephard (2006), Ou et al. (2023), it necessitates the iterative application of the QRNN cell to a sequence of input words $in_1, in_2, \ldots, in_L$. The outgoing lanes out_i mark a distinct distribution assessing p_i across the class labels.

3.12 Model Implementation and Training Details

Table 2 summarizes the structure and function of each hybrid NN–ARMA–GARCH model used in this study. It outlines the core architecture, the method of fractal integration, input design, and technical implementation details. All models were developed in Python 3.10. Training was conducted on a workstation equipped with an Intel Core i9-12900 K CPU, 128 GB RAM, and an NVIDIA RTX A6000 GPU (48 GB VRAM). We used the Adam optimizer with a learning rate of 0.001 and batch size of 128. Validation was carried out using both Nearest-K Cross Validation (NK-CV) and Random Cross Validation (Rand-CV) to ensure temporal consistency and generalizability. All models employed early stopping based on validation loss to prevent overfitting.

Table 2 Summary of F-NN–ARMA–GARCH models and implementation details

Model	Architecture	Fractal Integration	Input Handling	Software
ARMA-GARCH-NN	Shallow Feed-forward Neural Network	None (benchmark)	Flat input vector of lagged series	TensorFlow 2.13
ARMA-GARCH-F-DRCNN	Deep Recurrent Convolutional Neural Network	Multiscale temporal inputs processed via CNN-RNN layers	Inputs structured by time-scale blocks	TensorFlow 2.13
ARMA-GARCH-F-DNDT	Neural Decision Tree with soft binning	Decision nodes operate on scale-specific variable groupings	Inputs partitioned by scale, soft thresholds	TensorFlow+DNDT module
ARMA-GARCH-F-QNN	Quantum Neural Network	Qubit rotations weighted by fractal scale relevance (λ_n)	Encoded quantum amplitudes from multiscale data	Qiskit
ARMA-GARCH-F-QRNN	Quantum Recurrent NN with multiscale memory states	Hidden states organized as scale-specific subspaces	Multiscale sequences fed into quantum gates	Qiskit

4 Sample and Data

The sample collects 5-min prices from the Refinitiv Eikon database, choosing two different moments of the GameStop stock prices in the market, from January 11, 2021–December 31, 2021. The upward trend moment covers dates from January 11, 2021, to January 29, 2021; and the downward trend occurs from February 1, 2021, to February 11 2021. Furthermore, we evaluate our approaches over an extended duration, spanning from February 15, 2021, to December 31, 2021. The attributes under consideration can be categorised into two distinct groups: components from time series and technical indicators. A concise overview of the 13 variables utilised in our study is presented in Table 3. In line with Sun et al. (2019)'s methodology, for predictive purposes, we incorporate historical variables with time lags ranging from 1 to 60. As a result, the input dataset encompasses a total of 780 variables.

Innovations are computed by modifying. the 5-min returns of the S&P 500 with ARMA(1,1)-GARCH(1,1). $|z_t|$ and sign(z_t) could also contain information about the target; hence, they are extracted as potential features. For prediction purposes, only lagged time series elements are considered in the candidate feature set. The lowest offset is 1 and the highest offset is 60.

Technical indicators typically manifest as mathematical functions involving open (the initial price), close (the final price), high (the maximum price during a specified timeframe), low (the minimum price during a specified timeframe), volume, and volatility. These functions serve as representations of the microstructural characteristics inherent in the market.

$$RN = \frac{1}{4\log(2)}(high_t - low_t)$$

True Range (TR): $TR_t = High_t - Low_t$

Bar Information (BI): The bar information indicators utilize the data from the bar chart and are calculated as follows:

Table 3 Data descriptions

Categories	Input variables	Variable No
Time-series elements	The residue (σ) across lag intervals 1–60	001–060
	Conditional Mean (h) across lag intervals 1–60	061–120
	Innovation (u) across lag intervals 1–60	121–180
	Innovation Sign (u)	181–240
	Return (r) across lag intervals 1–60	241–300
	Return Sign across lag intervals 1–60	301–360
Technical indicators	Realized Variance across lag intervals 1–60	361–420
	Bi-power Variance across lag intervals 1–60	421–480
	Range across lag intervals 1–60	481–540
	True Range across lag intervals 1–60	541–600
	Bar Information I across lag intervals 1–60	601–660
	Bar Information II across lag intervals 1–60	661–720
	Bar Information III across lag intervals 1–60	721–780

$$BI1_t = (High_t - Open_t)/= (High_t - Low_t)$$

$$BI2_t = (Open_t - Close_t)/= (High_t - Low_t)$$

$$BI3_t = (Close_t - Low_t)/= (High_t - Low_t)$$

Realized Variance (rv): Based on n intra-day returns, the Realized Variance, rv, is calculated as $rv = \sum_{i=2}^{n} \left(p_{\frac{it}{n}} - p_{\frac{(i-1)t}{n}} \right)^2$, being $p_t = \log(P_t)$, and P_t the price at time t.

Bi-Power Volatility (rv′): Bi-power volatility is a measure of HF volatility in the presence of unusual jumps (Sun et al., 2019). It is calculated as $rv' = \sum_{i=3}^{n} \left| p_{\frac{it}{n}} - p_{\frac{(i-1)t}{n}} \right| \left| p_{\frac{it}{n}} - p_{\frac{(i-2)t}{n}} \right|$.

Regarding the input variables, we conduct a preliminary test using arbitrarily allocated samples to determine the optimal quantity of these variables. One-step-ahead predictions are performed across 100 moving windows, employing randomly selected samples distinct from those used in the primary model training and testing phases. Table 4 outlines the evolution of the root mean square error (RMSE) in connection with training and testing outcomes, considering selections of 10, 20, 30, and 40 variables. Figures 2, 3, 4, 5, and 6 illustrate the mean RMSE for each technique with varying input quantities. These charts depict the test results derived from the choices of 10, 20, 30, and 40 variables, utilizing the MRMR feature selection approach. Specifically, they showcase the fluctuations in root-mean-square-error (RMSE) based on the test set. Additionally, Annex A presents two figures for the MS-ARMA-GARCH-MLP and MS-ARMA-GARCH-RBF methods—figures 22 and 23, respectively.

While the model's fit to the training set improves with an increase in the number of inputs, there's a risk of overfitting. A shift in performance is observed when moving from 20 to 30 entries, as evidenced by the test set results. Similar tests based on CMIM also yield consistent outcomes. Consequently, we set 20 as the optimum number of input variables. Notably, the QNN method attains the best average RMSE in the testing set with 20 features (0.027873), followed by the QRNN approach (0.040254), with the DNDT technique securing the third position (0.045928).

5 Results

Employing the MRMR and CMIM feature selection algorithms alongside random cross-validation (Rand-CV) and k-nearest cross-validation (NK-CV), we scrutinize the predictive efficacy of our methodologies. MRMR operates by selecting a subset of features with the highest correlation to the class (output) and the lowest correlation among themselves. On the other hand, CMIM avoids selecting features already chosen, as they provide no additional information about the forecasted class. Thus, this criterion ensures a judicious balance between independence and discrimination. The prediction outputs are assessed based on the accuracy index for each method (see Table 5) and the RMSE, as illustrated in Table 6.

Table 4 Average RMSEs on training and test ensembles with the respective approaches

Features	Training	Testing
ARMA-GARCH-NN		
10	0.237832	0.247057
20	0.103627	0.224652
30	0.182359	0.198076
40	0.144763	0.183088
ARMA-GARCH-F-DCRNN		
10	0.175188	0.184596
20	0.189856	0.221009
30	0.168243	0.168851
40	0.129486	0.149201
ARMA-GARCH-F-DNDT		
10	0.061905	0.065467
20	0.064651	0.045928
30	0.041534	0.057266
40	0.042478	0.046936
ARMA-GARCH-F-QNN		
10	0.034782	0.036815
20	0.027745	0.027873
30	0.027944	0.035802
40	0.033291	0.040353
ARMA-GARCH-F-QRNN		
10	0.030717	0.033259
20	0.028527	0.040254
30	0.021014	0.040922
40	0.017392	0.034057

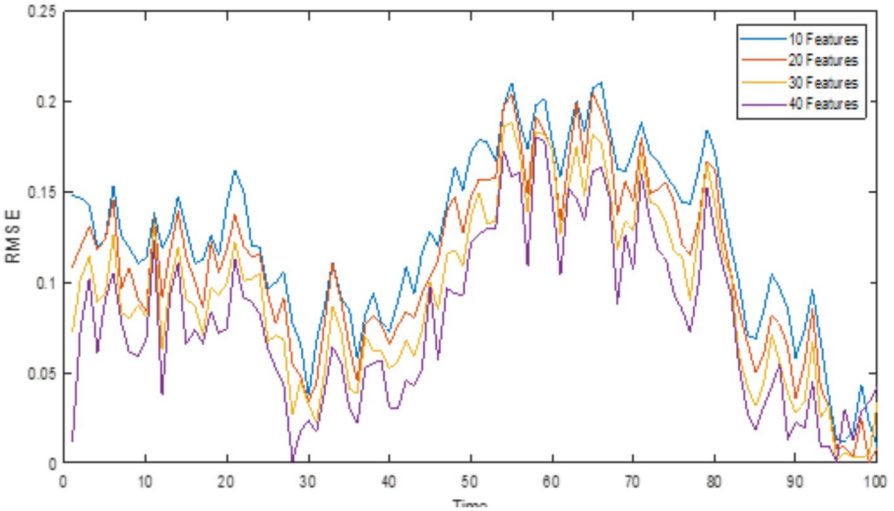

Fig. 2 RMSEs for varying numbers of selected features in ARMA-GARCH-NN (Testing sets)

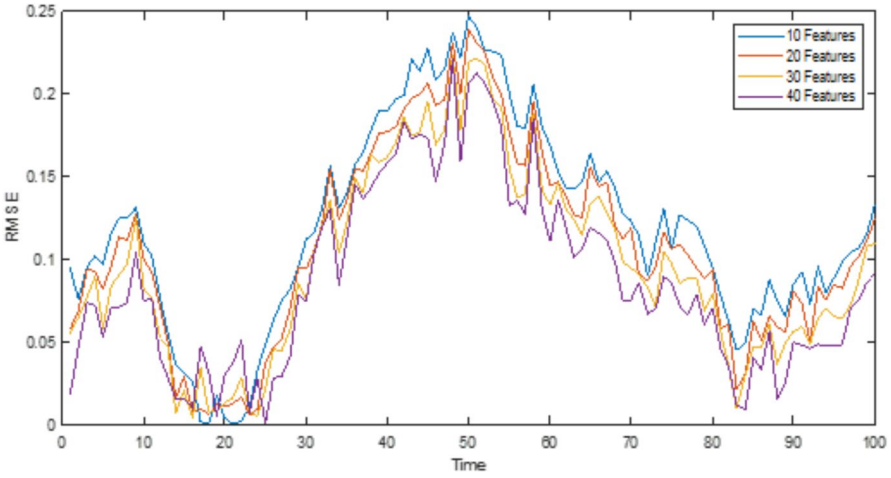

Fig. 3 RMSEs for varying numbers of selected features in ARMA-GARCH- F-DCRNN (Testing sets)

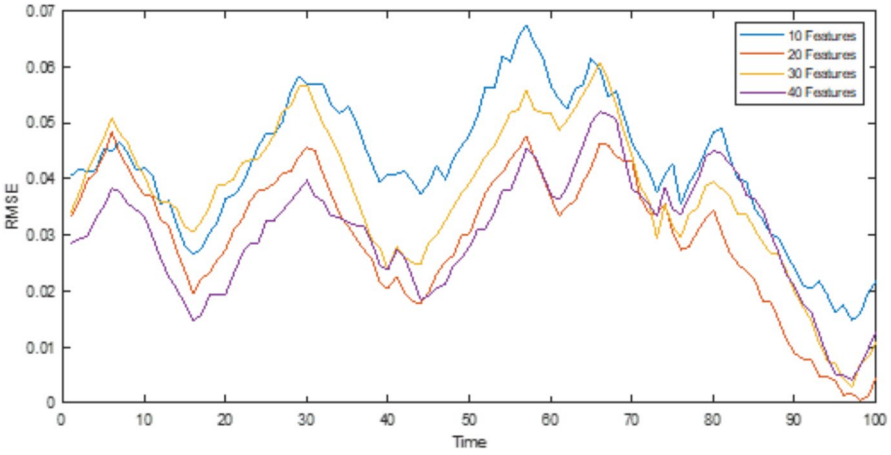

Fig. 4 RMSEs for varying numbers of selected features in ARMA-GARCH- F-DNDT (Testing sets)

It is apparent that the integration of MRMR with the NK-CV method significantly enhances the predictive accuracy across all techniques when compared to the use of Rand-CV. Notably, employing NK-CV yields superior outcomes for MRMR, whereas for CMIM, Rand-CV emerges as the most effective approach. Ultimately, across all methodologies, the results for root mean square error (RMSE) provide substantiation that CMIM outperforms MRMR, albeit the disparity in outcomes is not particularly pronounced.

We investigate the effectiveness of our devised thresholded ensemble voting technique in enhancing prediction performance (Table 7). In the ensemble voting setup, we establish the threshold at 5%. This configuration guides the prediction process as detailed below. To ensure model independence, we arbitrarily choose 90% of objects

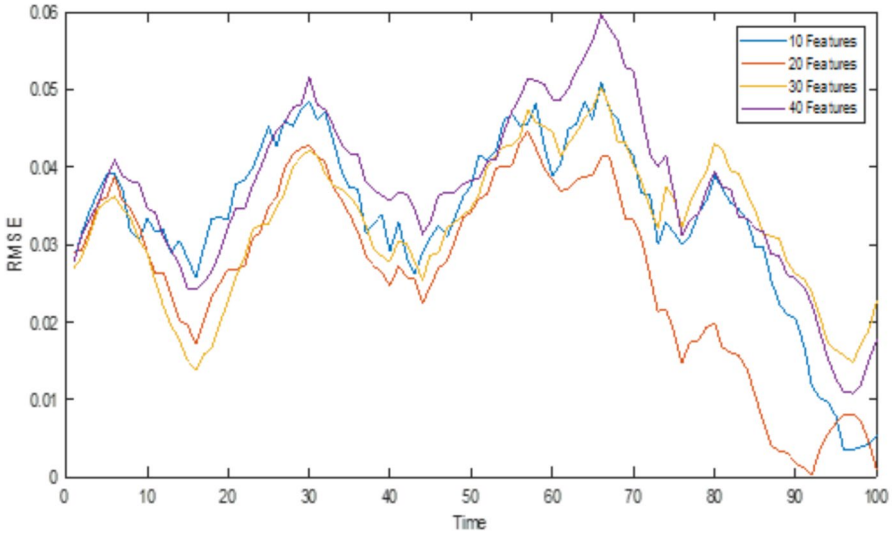

Fig. 5 RMSEs for varying numbers of selected features in ARMA-GARCH- F-QNN (Testing sets)

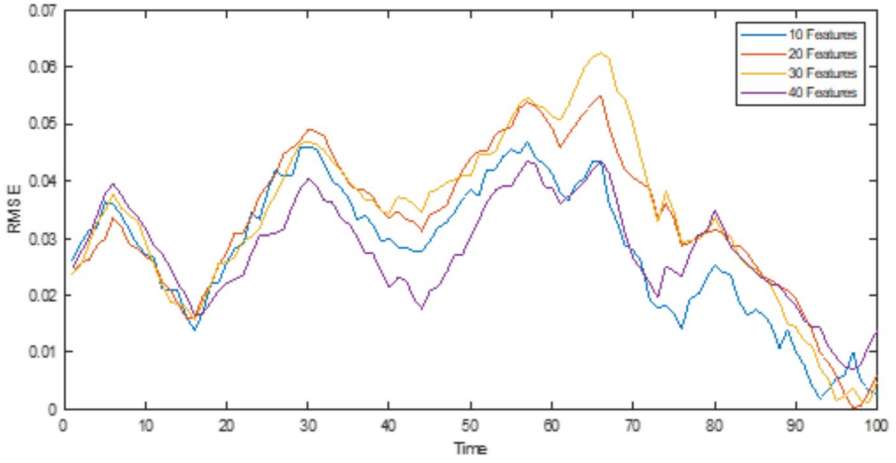

Fig. 6 RMSEs for varying numbers of selected features in ARMA-GARCH- F-QRNN (Testing sets)

from the original training set for each training set iteration. Additionally, the initial weights of neural networks are randomly assigned. Following training, in the 50 models, if the count of positive innovations exceeds 28, a positive market shock is predicted; if it falls below 22, a negative market shock is predicted. Instances where the count of positive innovations ranges from 22 to 28 are considered as lacking a strong signal indicating a significant upcoming innovation. Thus, no predictions are made in such scenarios, aligning with the methodology in some recent papers (Sun et al., 2019). Employing the CMIM-NK-CV model, accuracy rates experience improvement through the application of threshold adjustment in both bottom-up and

Table 5 Performance comparison in predicting future direction of market shocks

ARMA-GARCH-NN				
	MRMR		CMIM	
Trend	NK-CV	Rand-CV	NK-CV	Rand-CV
Upward	**0.519**	0.492	0.428	0.496
	0.007	0.215	0.132	0.034
Downward	0.473	0.431	0.457	0.508
	0.002	**0.348**	0.019	0.003
ARMA-GARCH-F-DCRNN				
	MRMR		CMIM	
Trend	NK-CV	Rand-CV	NK-CV	Rand-CV
Upward	**0.477**	0.453	0.398	0.454
	0.007	0.207	0.109	0.008
Downward	0.471	0.425	0.427	0.493
	0.015	0.308	0.014	0.040
ARMA-GARCH-F-DNDT				
	MRMR		CMIM	
Trend	NK-CV	Rand-CV	NK-CV	Rand-CV
Upward	**0.510**	0.479	0.405	0.457
	0.032	0.208	0.129	0.031
Downward	0.498	0.428	0.435	0.534
	0.026	0.333	0.043	0.012
ARMA-GARCH-F-QNN				
	MRMR		CMIM	
Trend	NK-CV	Rand-CV	NK-CV	Rand-CV
Upward	**0.554**	0.480	0.431	0.464
	0.048	0.253	0.173	0.076
Downward	0.527	0.473	0.448	0.555
	0.041	0.342	0.060	0.006
ARMA-GARCH-F-QRNN				
	MRMR		CMIM	
Trend	NK-CV	Rand-CV	NK-CV	Rand-CV
Upward	**0.551**	0.477	0.415	0.423
	0.041	0.253	0.134	0.043
Downward	0.485	0.435	0.417	0.552
	0.029	0.298	0.122	0.017

top-down samples. Table 7 demonstrates a systematic enhancement in performance for all methods when the threshold is integrated into the prediction procedure. For the uptrend, the accuracy rate raises to an average of 48.24% across all approaches; for the downtrend, the accuracy rate reaches an average of 54.84%. Both results surpass 48%, indicating precise predictions of market disturbances.

To ensure the robustness of our findings, we conduct a second round of tests employing a randomly selected sample spanning from February 15, 2021, to December 31, 2021. Figures 7, 8, 9, 10 and 11 show the outcomes of the MRMR-NK-CV model with and without a threshold for each methodology applied. Utilizing the MRMR-NK-CV model, with and without the threshold, we generate charts for each methodology, comparing their performance over the entire sample. In Annex B, two figures showcasing the MS-ARMA-GARCH-MLP and MS-ARMA-GARCH-RBF

Table 6 Forecast performance comparison of RMSE

ARMA-GARCH-NN				
	MRMR		CMIM	
Trend	NK-CV	Rand-CV	NK-CV	Rand-CV
Upward	0.834	0.794	0.760	**0.742**
Downward	0.871	0.842	0.832	**0.820**
ARMA-GARCH-F-DCRNN				
	MRMR		CMIM	
Trend	NK-CV	Rand-CV	NK-CV	Rand-CV
Upward	0.711	0.691	0.650	**0.649**
Downward	0.716	0.702	0.661	**0.643**
ARMA-GARCH-F-DNDT				
	MRMR		CMIM	
Trend	NK-CV	Rand-CV	NK-CV	Rand-CV
Upward	0.626	0.614	0.576	**0.534**
Downward	0.514	0.484	0.446	**0.404**
ARMA-GARCH-F-QNN				
	MRMR		CMIM	
Trend	NK-CV	Rand-CV	NK-CV	Rand-CV
Upward	0.453	0.411	0.388	**0.379**
Downward	0.428	0.414	0.379	**0.336**
ARMA-GARCH-F-QRNN				
	MRMR		CMIM	
Trend	NK-CV	Rand-CV	NK-CV	Rand-CV
Upward	0.324	0.293	0.277	**0.253**
Downward	0.194	0.166	0.128	**0.112**

methods are presented—Figures 24 and 25, respectively. Furthermore, we compare the average accuracy rate for each technique (see Table 8), with ARMA-GARCH-F-QRNN emerging as the top-performing method (91.87%), followed by ARMA-GARCH-F-QNNN (90.61%), and in the third position, ARMA-GARCH-F-DNDT (88.29%). This supports the efficacy of our model in forecasting market impacts.

Furthermore, we evaluate the efficacy of our hybrid model to optimize approaches for predicting market disturbances and integrating them into trading strategies. We introduce a novel trading signal in conjunction with predictions based on ARMA (1,1)-GARCH (1,1). The performance of the trading strategy is assessed using an arbitrarily chosen sample of 5-min stock prices. Figures 12, 13, 14, 15 and 16 provide a comparative analysis of the cumulative return derived from the trading strategy, considering signals from each approach with varying thresholds. When adjusting thresholds in Figures 12, 13, 14, 15 and 16, it is crucial to consider how changes in threshold values may impact the initiation of the forecasting process. According to the specified criterion (Eq. 8), the decision to initiate the forecasting process depends on the number of positive votes exceeding a specific threshold. Consequently, varying this threshold may influence when the system deems it appropriate to generate predictions based on perceived substantial innovation. In Annex C, two figures illustrating the MS-ARMA-GARCH-MLP and MS-ARMA-GARCH-RBF methods are provided—Figures 26 and 27, respectively. Our sample comprises 1,000 '5-min' time points, representing returns over approximately 14 trading days. The minimum profitability achieved is 1.8% in the ARMA-GARCH-NN method, followed by ARMA-

Table 7 Improved performance with joint voting (NK-CV) by precision index

ARMA-GARCH-NN		
Trend	With threshold	Without threshold
Upward	**0.510**	0.496
	0.006	0.134
Downward	**0.525**	0.508
	0.007	0.053
ARMA-GARCH-F-DCRNN		
Trend	With threshold	Without threshold
Upward	**0.455**	0.454
	0.028	0.058
Downward	**0.515**	0.493
	0.002	0.070
ARMA-GARCH-F-DNDT		
Trend	With threshold	Without threshold
Upward	**0.489**	0.457
	0.055	0.031
Downward	**0.542**	0.534
	0.082	0.013
ARMA-GARCH-F-QNN		
Trend	With threshold	Without threshold
Upward	**0.509**	0.464
	0.103	0.076
Downward	**0.568**	0.555
	0.011	0.006
ARMA-GARCH-F-QRNN		
Trend	With threshold	Without threshold
Upward	**0.449**	0.423
	0.008	0.094
Downward	**0.592**	0.552
	0.003	0.105

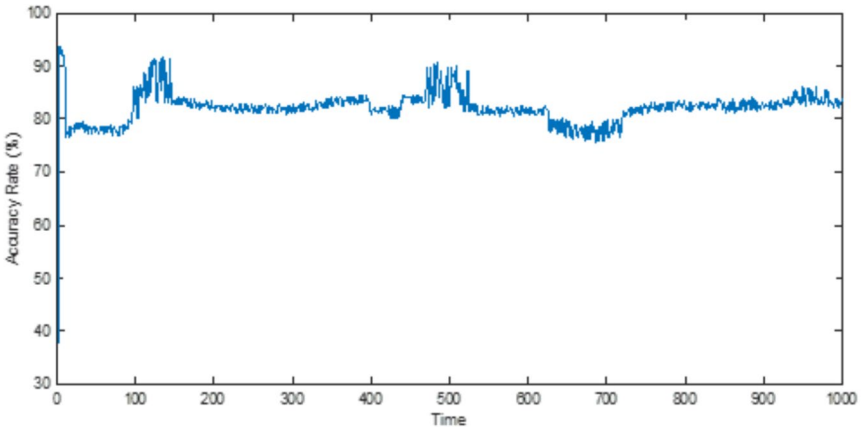

Fig. 7 Comparison of results on a long-term sample using ARMA-GARCH-NN

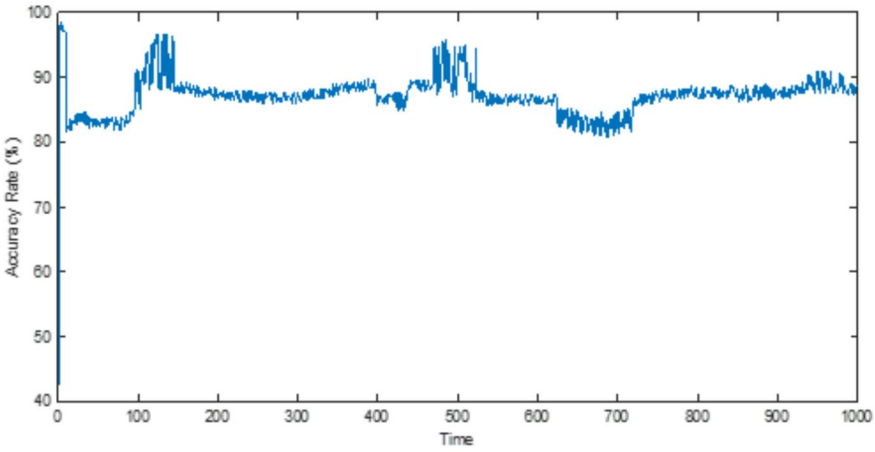

Fig. 8 Comparison of results on a long term sample basisusing ARMA-GARCH-F-DCRNN

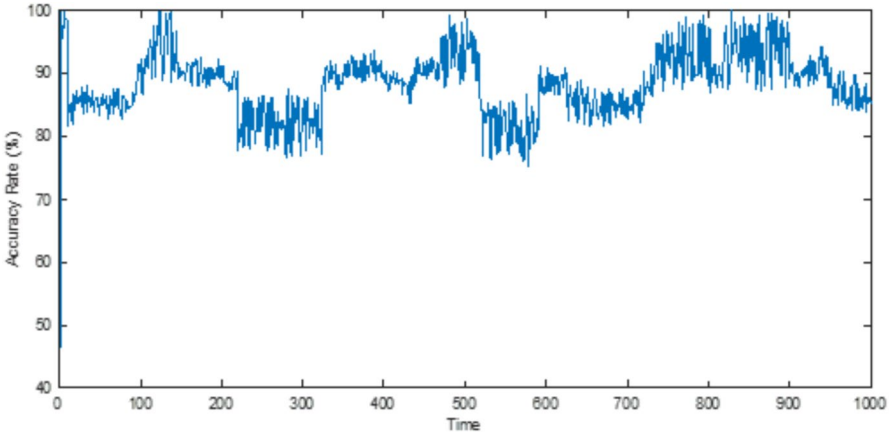

Fig. 9 Comparison of results on a long term sample basis using ARMA-GARCH-F-DNDT

GARCH-F-DCRNN (1.9%), while the highest profitability reaches 3.6% with the ARMA-GARCH-F-QRNN method, closely followed by both ARMA-GARCH-QNN (3.4%) and ARMA-GARCH-F-DNDT (3.1%). Remarkably, the ARMA-GARCH-F-QRNN and ARMA-GARCH-F-QNN techniques demonstrate significant improvement over other methods in terms of cumulative returns.

Furthermore, in Figs. 17, 18, 19, 20 and 21, we present the cumulative performance trends over time by establishing a threshold at 0.00025. For these Figs. 17, 18, 19, 20 and 21, the cumulative performance trends unfold within the framework of a meticulously chosen threshold set at 0.00025, a pivotal parameter in our forecasting methodology. This threshold intricately influences the initiation of the forecasting process, determined by an initiation criterion that evaluates the number of positive votes from the uppermost v% of models. In Annex D, two figures portraying the

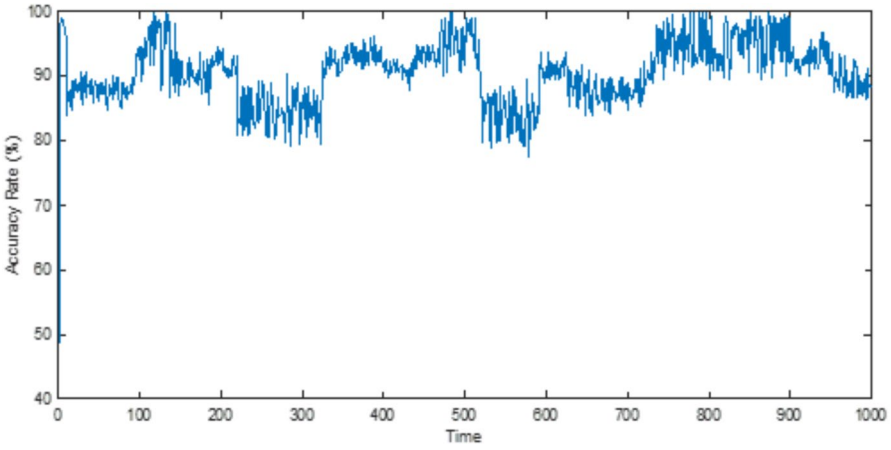

Fig. 10 Comparison of results on a long term sample basis using ARMA-GARCH-F-QNN

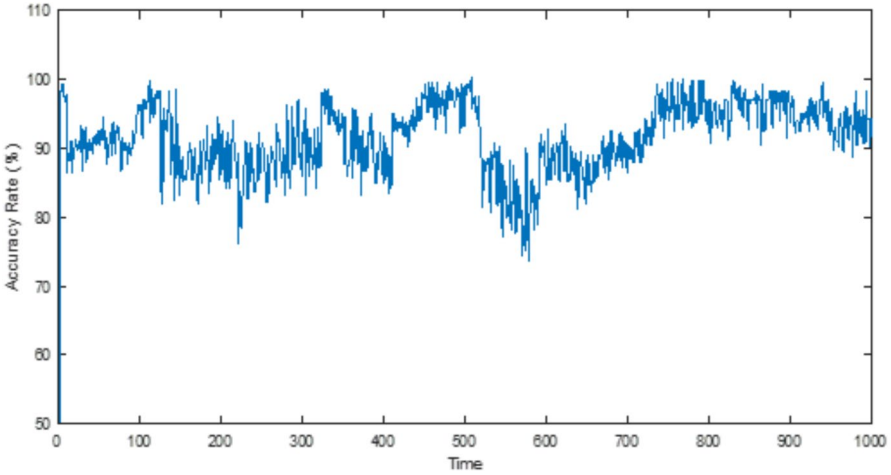

Fig. 11 Comparison of results on a long term sample basis using ARMA-GARCH-F-QRNN

Table 8 Comparison of average performance on a long-term sample basis

	Average (%)
ARMA-GARCH-NN	81.10
ARMA-GARCH-F-DCRNN	87.16
ARMA-GARCH-F-DNDT	88.29
ARMA-GARCH-F-QNN	90.61
ARMA-GARCH-F-QRNN	91.87

MS-ARMA-GARCH-MLP and MS-ARMA-GARCH-RBF methods are included—Figures 28 and 29, respectively. Once again, we demonstrate the persistent outperformance of ARMA-GARCH-F-QRNN and ARMA-GARCH-F-QNN, consistently

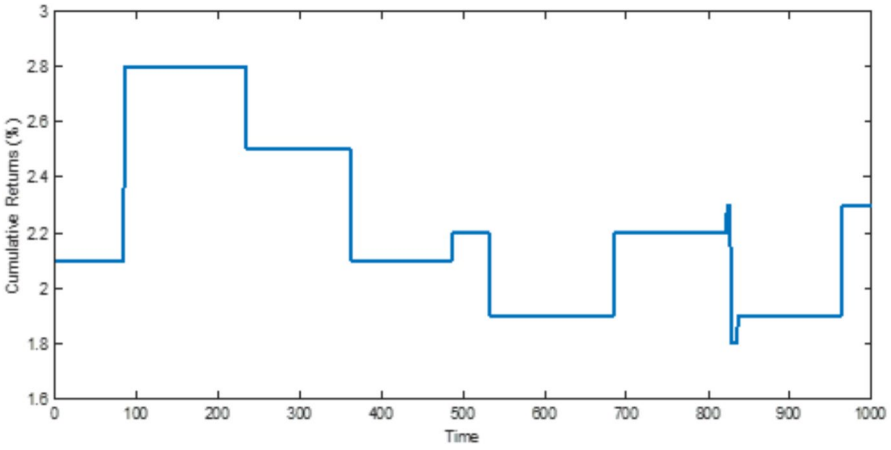

Fig. 12 Cumulative Return under Different Thresholds using ARMA-GARCH-NN

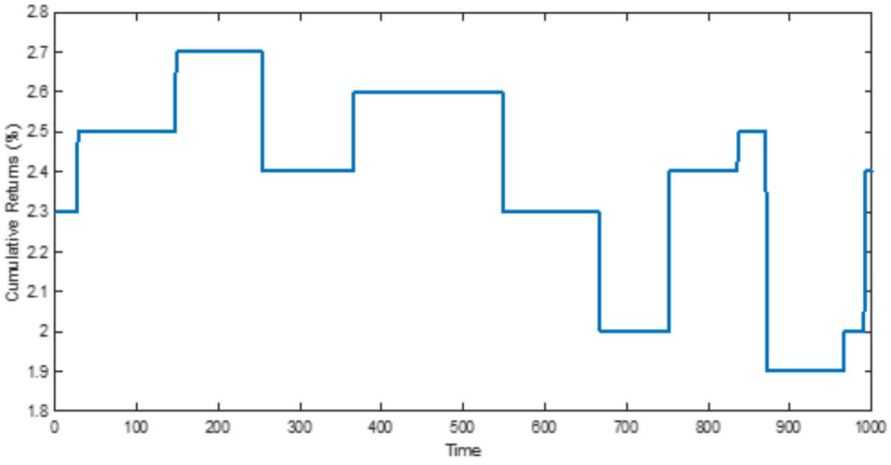

Fig. 13 Cumulative Return under Different Thresholds using ARMA-GARCH-F-DCRNN

achieving maximum returns of 3.6% and 3.4%, respectively, surpassing the performance of other techniques.

In conclusion, Table 9 details the accuracy levels attained by each method during both the training and testing phases. The testing data is employed to assess the constructed model and make predictions. Across all instances, the accuracy levels consistently surpass 73.17%. ARMA-GARCH-QRNN emerges as the method with the highest accuracy level (91.28%). Following closely is ARMA-GARCH-F-QNN, succeeded by ARMA-GARCH-F-DNDT, and finally ARMA-GARCH-F-DCRNN. It's noteworthy that, as indicated in Table 9, all the techniques achieve commendably high precision levels.

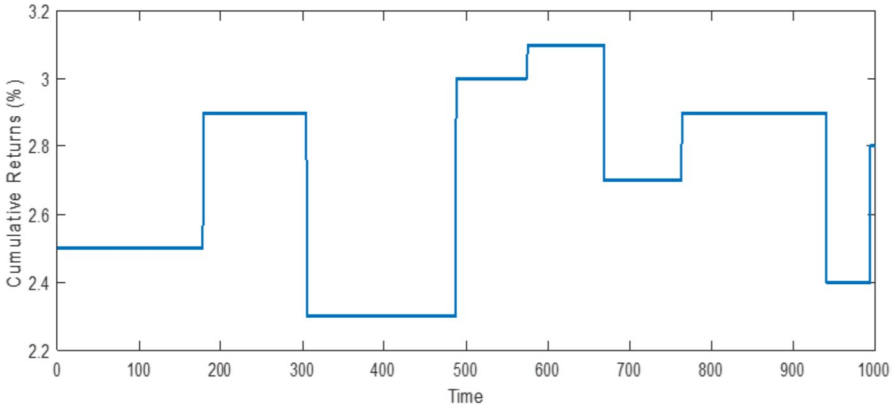

Fig. 14 Cumulative Return under Different Thresholds using ARMA-GARCH-F-DNDT

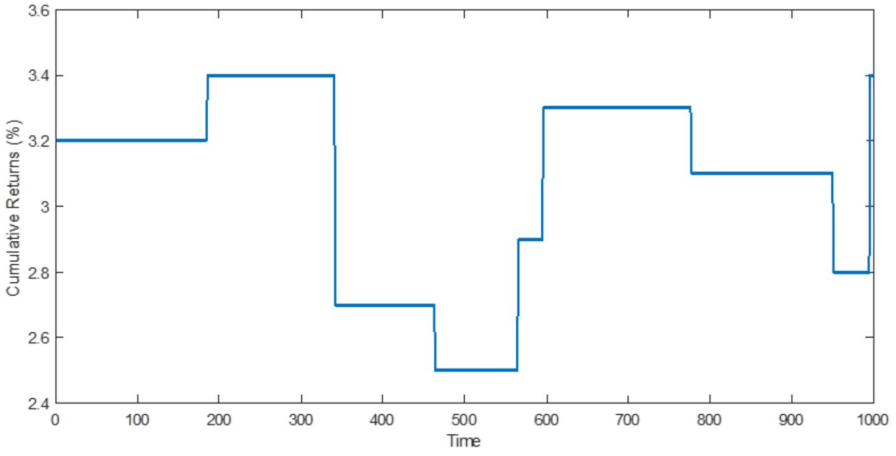

Fig. 15 Cumulative Return under Different Thresholds using ARMA-GARCH-F-QNN

6 Discussion of Results

In the context of stock market forecasting, our research analyzes the short squeeze phenomenon over the period January 11 to December 31, 2021. We propose an innovative model that integrates fractal properties within a hybrid machine learning framework. The ARMA-GARCHG-F-QRNNN model is the one that achieves a better accuracy rate (91.28%), far outperforming traditional and contemporary models. In terms of the RMSE results, all models corroborate that the CMIM outperforms the MRMR, although the disparity in results is not particularly pronounced. For our ARMA-GARCH-F-QRNN model, the Rand-CV methodology has a higher performance than the NK-CV to evaluate the performance of machine learning models, being 0.253 in uptrends and 0.112 in downtrends.

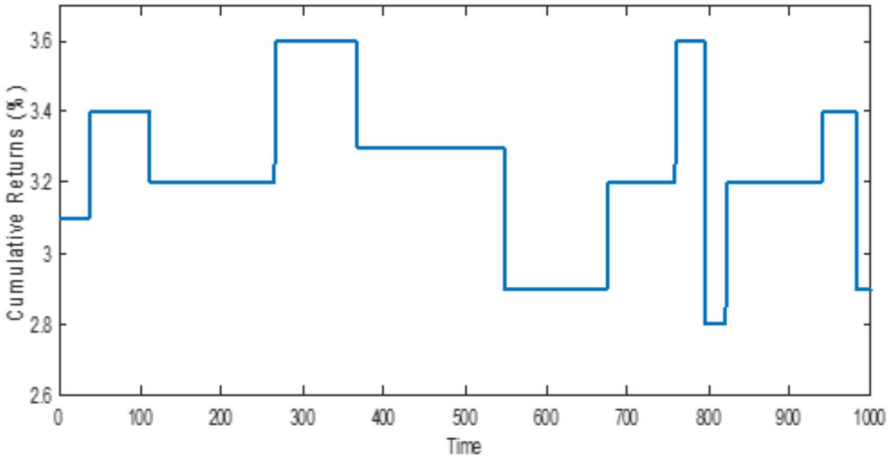

Fig. 16 Cumulative Return under Different Thresholds using ARMA-GARCH-F-QRNN

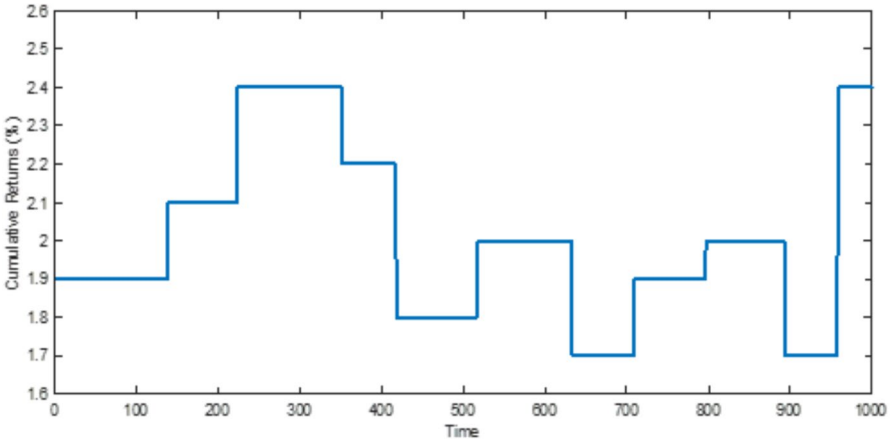

Fig. 17 Cumulative Returns (threshold = 0.00025) using ARMA-GARCH-NN

In contrast to previous investigations utilizing GARCH models, Naseem et al. (2018) explore the volatility of the Pakistani stock market over the period from January 1, 2008, to June 30, 2018. Their analysis involves various GARCH-type models, encompassing both symmetric (GARCH & GARCH-M) and asymmetric (EGARCH & TGARCH) variants. Notably, the GARCH-M (1, 1) model exhibits significant positivity at the 1% level in the Std. and GED scores, indicating the presence of a risk premium, although other parts of the distribution show insignificance. However, this study does not provide clear metrics of precision or RMSE that can be directly comparable. Meanwhile, Salamat et al. (2020) formulate a model based on symmetric (GARCH 1,1) and asymmetric (EGARCH, TGARCH, PGARCH) models from the GARCH family to study cryptocurrency volatility. The findings emphasize PGARCH as the best-fitted model, highlighting the substantial impact of positive

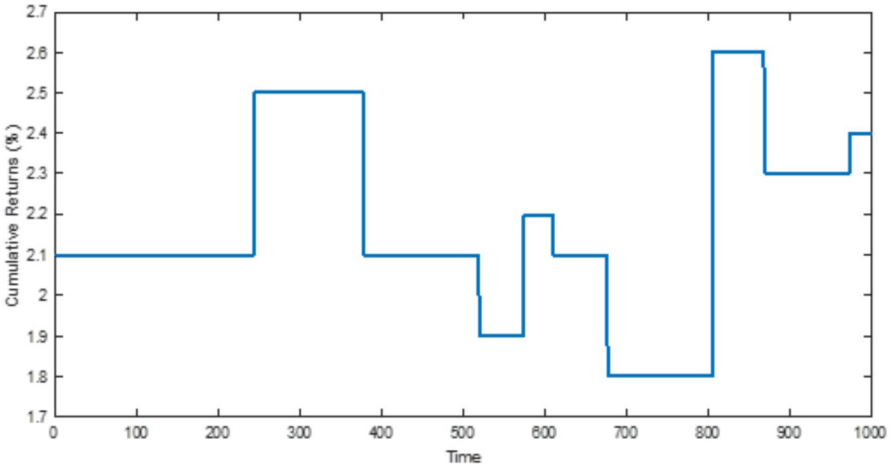

Fig. 18 Cumulative Returns (threshold = 0.00025) using ARMA-GARCH-F-DRCNN

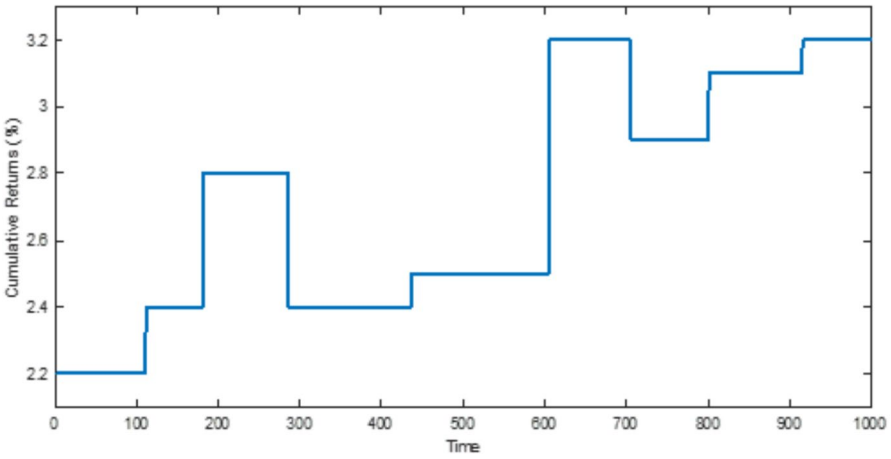

Fig. 19 Cumulative Returns (threshold = 0.00025) using ARMA-GARCH-F-DNDT

shocks on conditional volatility compared to negative shocks. However, his approach focuses more on the description of volatility dynamics than on quantitative predictive accuracy, which limits his direct comparability with deep learning models. In a different context, Arashi and Rounaghi (2022) employ the ARMA-GARCH model to forecast the daily return series of the NASDAQ stock exchange, demonstrating its effective forecasting capabilities with a remarkable 1% error level. However, when compared to the RMSE of our ARMA-GARCHG-F-QRNNN model, superior performance is evident, especially when considering the complexity and variability of the intraday data on which it was trained.

Shifting focus to Wu et al. (2022), experiments using real-world market data from the Chinese market index (January 1, 2010, to December 31, 2019) reveal that ARMA

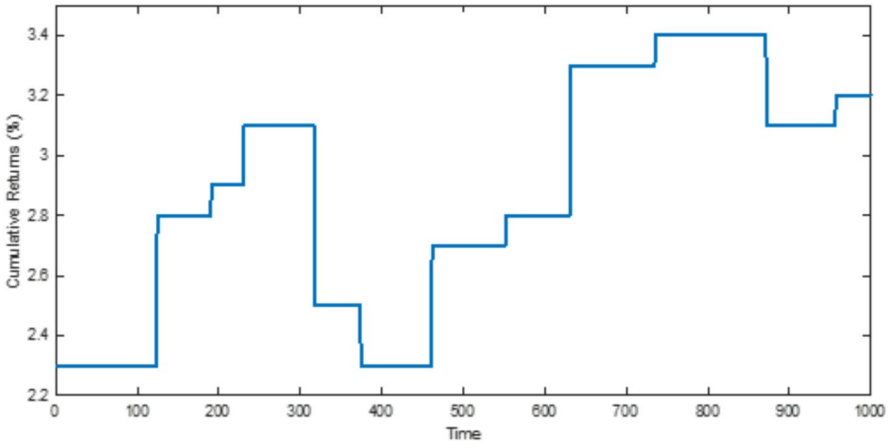

Fig. 20 Cumulative Returns (threshold = 0.00025) using ARMA-GARCH-F-QNN

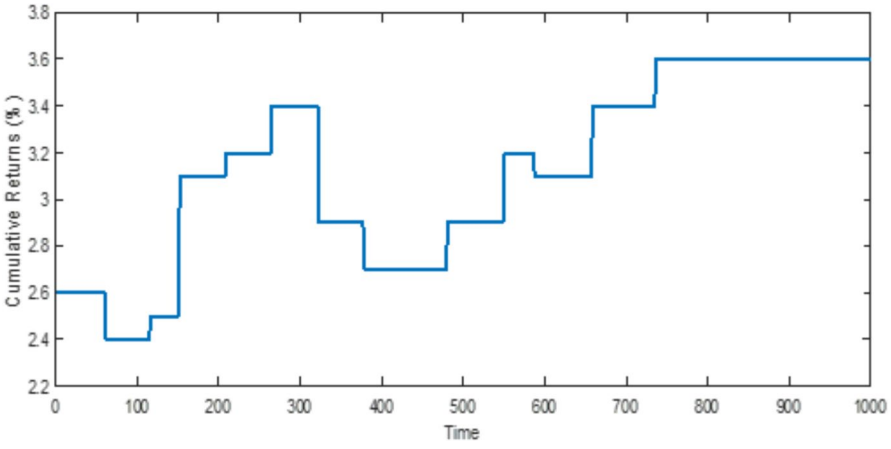

Fig. 21 Cumulative Returns (threshold = 0.00025) using ARMA-GARCH-F-QRNN

Table 9 Results of Accuracy

	Training	Testing
ARMA-GARCH-NN	77.89	73.17
ARMA-GARCH-F-DCRNN	85.67	84.72
ARMA-GARCH-F-DNDT	87.66	86.54
ARMA-GARCH-F-QNN	90.39	89.42
ARMA-GARCH-F-QRNN	93.76	91.28

and GARCH models fall short in competitive performance when compared to the deep learning method based on LSTM. Despite the Chinese stock market's preference for high precision and recall, especially for limited stock predictions, maintaining consistently high precision is deemed crucial for the profitability of their framework. Finally, Syuhada et al. (2023) conclude that GARCH (TGARCH), exponential

GARCH (EGARCH), asymmetric power ARCH (APARCH), and their fractionally integrated variations (e.g., FIGARCH, FIEGARCH, and FIAPARCH) possess the capacity to capture more intricate empirical characteristics of asset returns and their volatility.

When comparing methods employed in the literature for the GME short squeeze event, Kim et al. (2023) utilize statistical methods, specifically employing linear regression, to analyze the impact of social sentiment among investors. Their findings suggest that, for the most part, individual investors followed the trading patterns proposed by the authors through their engagement on social media platforms during the GME short squeeze. Moreover, their study indicates that social information from Reddit had a more pronounced effect on the trading activity of GME stock during the short squeeze compared to the influence exerted by Twitter. In a different approach, Hilliard and Hilliard (2023) apply regression models to investigate potential violations of the no-arbitrage condition in the context of the GME stock squeeze, identifying violations in both pre-squeeze and squeeze periods. Their main conclusion emphasizes that violations are more prevalent in longer maturities, and the market predominantly adhered to friction-adjusted no-arbitrage conditions during the GME squeeze, suggesting rational behavior. Meanwhile, Vaughan et al. (2023) combine non-negative matrix factorization (NMF) thematic models with manual content analysis techniques to explore the GameStop short squeeze across three online spaces: the WallStreetBets subreddit, the #GameStop hashtag on Twitter, and relevant digital news sources in the United States. Their findings lead to the conclusion that digital platforms have the capacity to establish boundaries and points of intermediation in the realm of contentious politics.

Andreev et al. (2022) conduct a comprehensive comparison with other methods, employing both non-autoregressive and autoregressive models to forecast the stock's returns for one, five, and seven days in the future. Their findings highlight that the most effective classifier is an autoregressive Random Forest model, boasting an impressive accuracy rate of 70%. This figure, although significant, is still considerably below the performance of our ARMA-GARCHG-F-QRNNN model, highlighting the impact of incorporating fractal structure and deep learning in the modeling of financial time series. In a 2023 study, Wang et al. (2023) introduce the"Primary Ensemble Empirical Mode Decomposition combined with Quantum Neural Network"(PEEMD-QNN) model, showcasing superior performance in forecasting Chinese stock index time series compared to other methods like the Back Propagation Neural Network, QNN model, and Ensemble Empirical Mode-Quantum Neural Network (EMD-QNN) model. This work does not provide concrete accuracy figures for comparison with our model. Other researchers explore hybrid approaches, as seen in the work of Cao et al. (2023), who presents an innovative hybrid quantum computing framework utilizing the Quantum Long Short-Term Memory (QLSTM) model for carbon price forecasting. Their model, the Linear-layer-enhanced Quantum Long Short-Term Memory (L-QLSTM), incorporates linear layers before and after the variational quantum circuits of QLSTM, leading to significantly enhanced learning accuracy compared to the standard QLSTM method. This study research does not report specific accuracy figures available for comparing with our model. Additionally, Sun et al. (2019) develop an ARMA-GARCH-NN method for intra-day stock

market shock forecasting, with results confirming its effectiveness in recognizing patterns within extensive stock data without relying on strong assumptions about distribution. This ARMA-GARCH-NN method, proven effective by Sun et al. (2019), is also applied in our model for comparative analysis. The previous studies do not provide concrete accuracy findings for comparison with our model. The latter study also does not yield accurate figures for comparison with our model.

In summary, our study provides an analysis of the short squeeze event on intraday market data with an ARMA-GARCH process, but also using different ML methods, not employed in prior work specifically related to the GME short squeeze event, achieving high levels of accuracy in all applied methods. Besides, our research shows how the integration of fractal properties and advanced neural network architectures can significantly improve the ability of predictive models to capture complex market patterns. The ARMA-GARCHG-F-QRNN model is thus positioned as a model proposal in the field of financial forecasting, especially useful in contexts of high volatility and nonlinear intraday dynamics.

7 Conclusions

This work has introduced a conceptual model that examines the short squeeze event on intraday market data using fractal dynamics integrated into an ARMA-GARCH framework. Our novel approach incorporates Deep Learning and Quantum methodologies, enhanced with fractal properties, across multiple models: ARMA-GARCH-NN, MS-ARMA-GARCH-MLP, MS-ARMA-GARCH-RBF, ARMA-GARCH-DRCNN, ARMA-GARCH-DNDT, ARMA-GARCH-QNN, and ARMA-GARCH-QRNN. The sample analyzed consists of GameStop short squeeze 5-min prices from January 11, 2021, to December 31, 2021. Our findings demonstrate the predictability of ARMA-GARCH short squeeze events at the intraday level, showcasing the effectiveness of fractal-enhanced methodologies.

The study highlights that the MRMR feature selection method performs notably better in forecasting short squeeze directions across all approaches, particularly when combined with NK-CV, which outperforms Rand-CV in MRMR. Conversely, in CMIM, Rand-CV delivers optimal performance. RMSE results indicate a slight advantage of CMIM over MRMR across all methods, though the differences are minor. Incorporating a threshold into the prediction process systematically improves performance across all approaches. Among these, the ARMA-GARCH-F-QRNN emerges as the top-performing technique, achieving the highest accuracy rates due to its multiscale and hierarchical modeling capabilities.

Unlike previous studies, our model significantly enhances accuracy rates by introducing threshold adjustments in both preceding and subsequent samples. Additionally, we employ neural network methodologies enriched with fractal dynamics, comparing their performance to traditional methods. The ARMA-GARCH-F-QRNN stands out for its superior adaptability and accuracy, demonstrating the practical advantages of integrating fractal principles into predictive modeling. These findings not only reaffirm the value of hybrid methods in volatile financial scenarios, but also provide an opening for new approaches in which fractal theory can enrich artificial

intelligence applied to markets. Indeed, the remarkable aspect is how fractal dynamics tools or quantum models can be translated into tangible improvements in stock market prediction.

Our study provides a valuable tool for practitioners to detect signs of market abnormalities during short squeezes. From a regulatory perspective, the findings suggest that financial regulators should consider integrating high-frequency analytics and machine learning tools into their market surveillance mechanisms. Early detection of short squeeze dynamics or unusual trading patterns could help mitigate systemic risks and improve transparency. Consequently, our research holds practical significance for both scholars and practitioners, aiding in the formulation of profitable investment strategies in markets affected by anomalies. For regulators and policymakers, the study emphasizes the need for continuous monitoring of investor groups on social media platforms, as their coordinated activities can disrupt market efficiency and potentially lead to crises. Furthermore, regulators should consider measures to address excessive short-selling, while acknowledging the constructive role of short sellers in identifying overvalued or distressed companies, which contributes to the overall health of financial markets..

In summary, this research makes a significant contribution to the field of finance by advancing trading automation and assisting institutional and retail investors in making informed decisions. The unique case of GameStop underscores the necessity for regulators to safeguard the integrity and stability of financial markets in the face of coordinated actions by investor groups.

Future research should explore the application of these fractal-enhanced techniques and trading strategies to other critical moments in financial markets, such as cryptocurrency bubbles, short squeezes in micro-cap stocks or fixed-income securities, sudden collapses or collective manipulation events. These scenarios would provide opportunities to test the robustness and adaptability of our methodologies in managing volatility and stabilizing trading strategies. Furthermore, the integration of non-price-based variables into GARCH modeling—such as trends in Reddit forums or the volume of Google searches—could further enhance the accuracy and utility of these models by incorporating qualitative information, offering deeper insights into market dynamics. Another future line of research might be the comparison of different markets (Europe, Asia, USA), which would allow validation of the robustness of the proposed approach in different contexts.

Appendix A

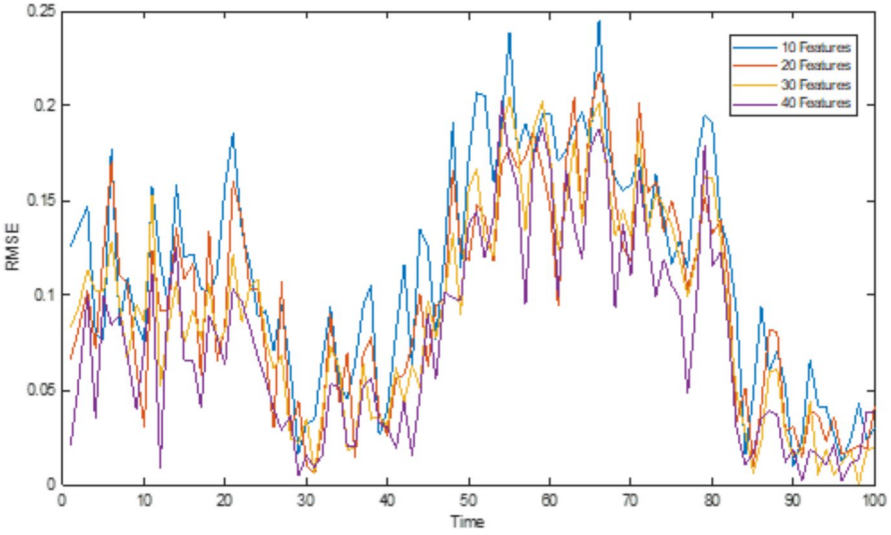

Fig. 22 RMSEs for varying numbers of selected features in MS-ARMA-GARCH-MLP (Test sets)

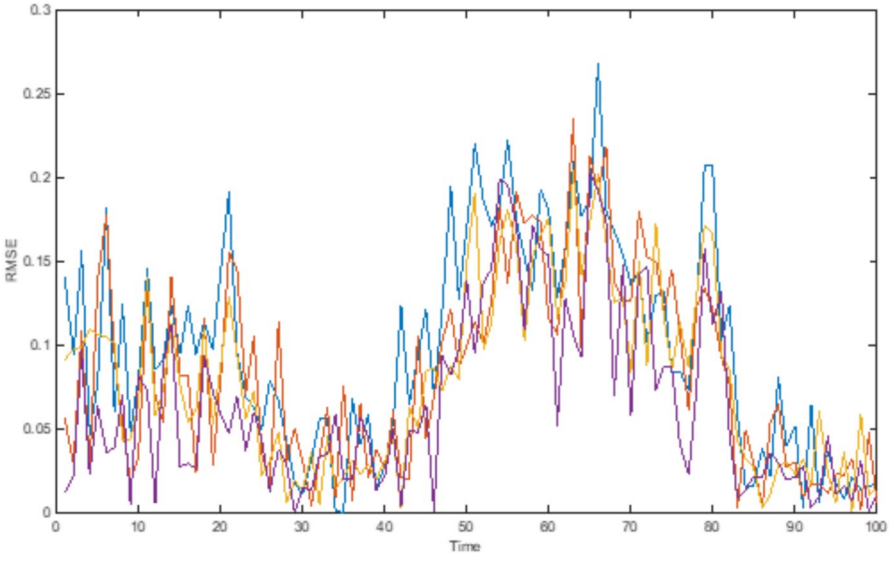

Fig. 23 RMSEs for varying numbers of selected features in MS-ARMA-GARCH-RBF (Test sets)

Appendix B

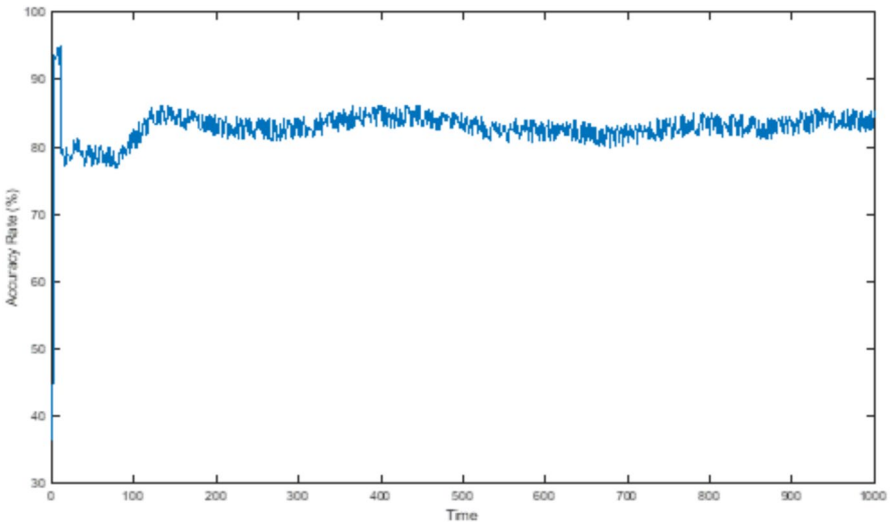

Fig. 24 Performance comparison based on a long-period sample using MS-ARMA-GARCH-MLP

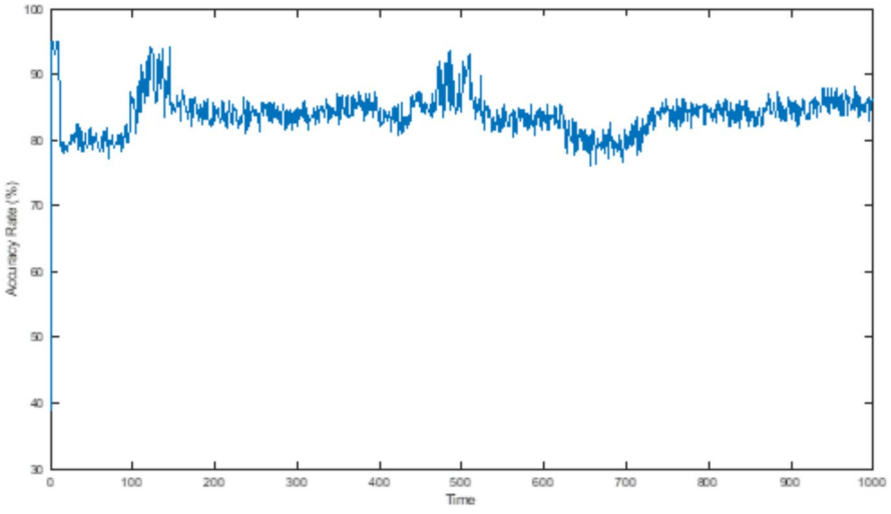

Fig. 25 Performance comparison based on a long-period sample using MS-ARMA-GARCH-RBF

Appendix C

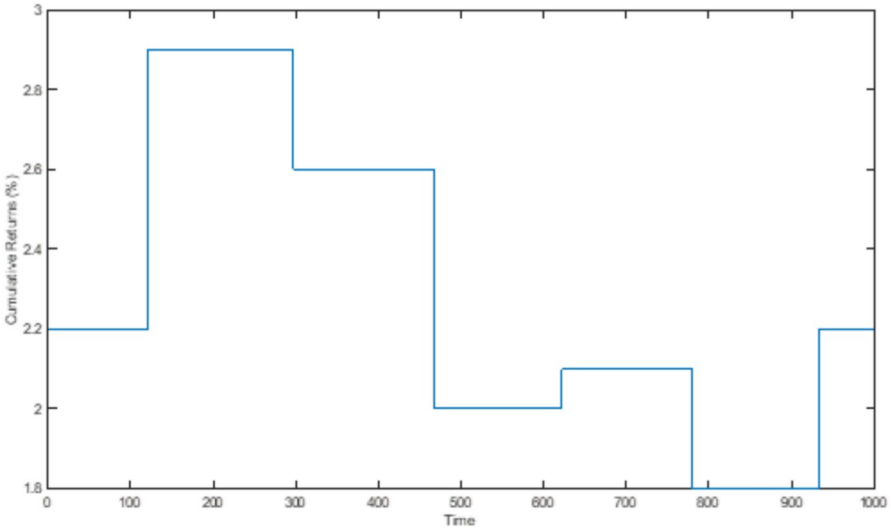

Fig. 26 Cumulative Return under Different Thresholds using MS-ARMA-GARCH-MLP

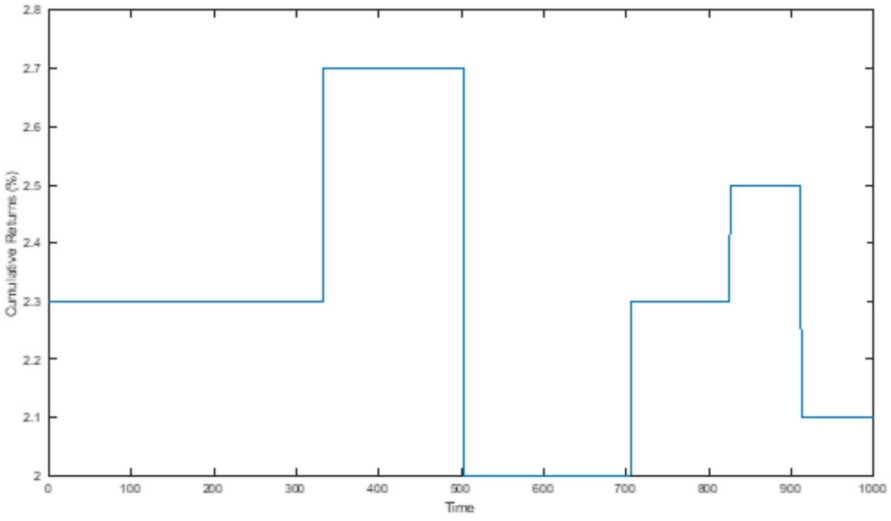

Fig. 27 Cumulative Return under Different Thresholds using MS-ARMA-GARCH-RBF

Appendix D

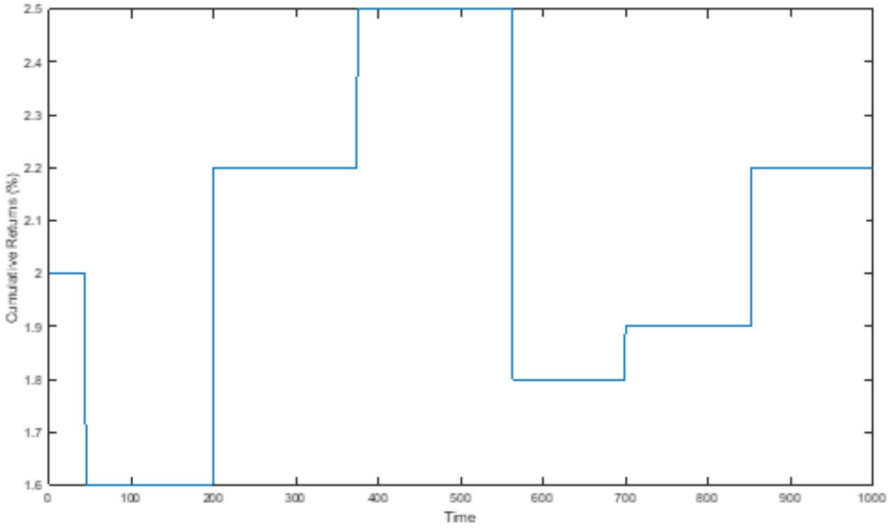

Fig. 28 Cumulative Returns (threshold = 0.00025) using MS-ARMA-GARCH-MLP

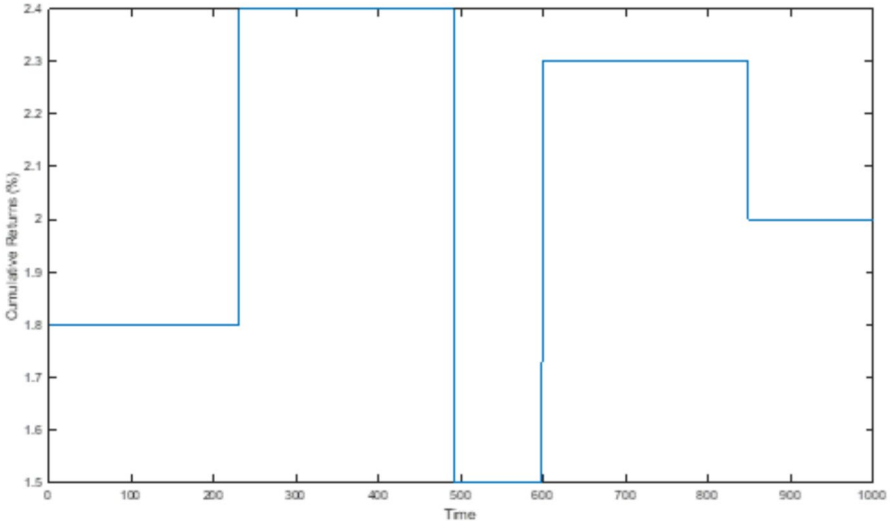

Fig. 29 Cumulative Returns (threshold = 0.00025) using MS-ARMA-GARCH-RBF

Authors'Contributions All authors have contributed equally to the conception, design, data analysis, interpretation of results, writing, and revision of the manuscript. All authors read and approved the final manuscript.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. Nature. No funding was received for the research, authorship, or publication of this manuscript.

Data Availability The datasets analyzed during the current study are available from the corresponding author upon reasonable request.

Declarations

Ethics Approval and Consent to Participate Not applicable. This manuscript does not involve studies on human participants, human data, human tissue, animals, or plants.

Consent for Publication Not applicable. This manuscript does not contain any individual person's data in any form.

Competing interests The authors declare that they have no competing financial or non-financial interests.

References

Akgiray V. (1989). Conditional heteroscedasticity in time series of stock returns: Evidence and forecasts. *Journal of Business*, 55–80.

Alaminos, D., Fernández, S. M., Neves, P. M., & Santos, J. A. C. (2019). Predicting sovereign debt crises with fuzzy decision trees. *Journal of Scientific & Industrial Research, 78*(11), 733–737.

Alaminos, D., Salas, M. B., & Fernández-Gámez, M. Á. (2023). Quantum Monte Carlo simulations for estimating FOREX markets: A speculative attacks experience. *Humanities and Social Sciences Communications, 10*(1), 353.

Alaminos, D., Salas, M. B., & Fernández-Gámez, M. A. (2024). Can Bitcoin trigger speculative pressures on the US Dollar? A novel ARIMA-EGARCH-Wavelet Neural Networks. *Physica A: Statistical Mechanics and its Applications, 654*, 130140.

Alaminos, D., Salas, M. B., & Fernández-Gámez, M. A. (2024b). High-Frequency Trading in Bond Returns: A Comparison Across Alternative Methods and Fixed-Income Markets. *Computional Economics, 64*, 2263–2354.

Alaminos, D., Salas, M. B., & Fernández-Gámez, M. A. (2024c). Global patterns and extreme events in sovereign risk premia: A fuzzy vs deep learning comparative. *Technological and Economic Development of Economy, 30*(3), 753–782.

Alaminos, D., Salas, M. B., & Fernández-Gámez, M. Á. (2024d). Hybrid genetic algorithms in agent-based artificial market model for simulating fan tokens trading. *Engineering Applications of Artificial Intelligence, 131*, Article 107713.

Alaminos, D., Salas, M. B., & Partal-Ureña, A. (2024e). Hybrid ARMA-GARCH-Neural Networks for intraday strategy exploration in high-frequency trading. *Pattern Recognition, 148*, Article 110139.

Allen, F., Haas, M. D., Nowak, E., & Tengulov, A. (2021). Market efficiency and limits to arbitrage: Evidence from the Volkswagen short squeeze. *Journal of Financial Economics, 142*(1), 166–194.

Andreev, B., Sermpinis, G., & Stasinakis, C. (2022). Modelling Financial Markets during Times of Extreme Volatility: Evidence from the GameStop Short Squeeze. *Forecasting, 4*(3), 654–673.

Arashi, M., & Rounaghi, M. M. (2022). Analysis of market efficiency and fractal feature of NASDAQ stock exchange: Time series modeling and forecasting of stock index using ARMA-GARCH model. *Future Business Journal, 8*(1), 1–12.

Bangyal WH, Qasim R, Rehman NU, Ahmad Z, Dar H, Rukhsar et al (2021). Detection of fake news text classification on COVID-19 using deep learning approaches, Computational and mathematical methods in medicine, 2021, 1–14.

Barndorff-Nielsen, O. E., & Shephard, N. (2006). Econometrics of testing for jumps in financial economics using bipower variation. *Journal of Financial Econometrics, 4*(1), 1–30.

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks, 5*(4), 537–550.

Bausch, J. (2020). Recurrent quantum neural networks. *Advances in Neural Information Processing Systems, 33*, 1368–1379.

Beer, K., Bondarenko, D., Farrelly, T., Osborne, T. J., Salzmann, R., Scheiermann, D., & Wolf, R. (2020). Training deep quantum neural networks. *Nature Communications, 11*(1), 808.

Benedetti, M., Lloyd, E., Sack, S., & Fiorentini, M. (2019). Parameterized quantum circuits as machine learning models. *Quantum Science and Technology, 4*(4), 043001.

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. *Nature, 549*(7671), 195–202.

Bildirici, M., & Ersin, O. (2014). Modeling Markov switching ARMA-GARCH neural networks models and an application to forecasting stock returns. *Scientific World Journal, 2014*(1), 497941.

Bui, Q., & Ślepaczuk, R. (2022). Applying hurst exponent in pair trading strategies on nasdaq 100 index. *Physica A: Statistical Mechanics and its Applications, 592*, 126784.

Cao, Y., Zhou, X., Fei, X., Zhao, H., et al. (2023). Linear-layer-enhanced quantum long short-term memory for carbon price forecasting. *Quantum Machine Intelligence, 5*(2), 26.

Cao Y, Guerreschi GG, Aspuru-Guzik A. (2017). Quantum neuron: an elementary building block for machine learning on quantum computers. https://arxiv.org/abs/1711.11240

Cerezo, M., Verdon, G., Huang, H. Y., Cincio, L., & Coles, P. J. (2022). Challenges and opportunities in quantum machine learning. *Nature Computational Science, 2*(9), 567–576.

Charef, F. (2023). Exchange Rate Forecasting: Nonlinear GARCH-NN Modeling Approach. *Annals of Data Science, 11*(3), 947–57.

Chohan UW. (2021). Counter-hegemonic finance: The gamestop short squeeze. Available at SSRN 3775127

Choy J, Wang B, AlShelahi A, Saigal R. (2021). Investor Impatience and Financial Markets: the Case of the Short Squeeze of Meme Stocks. Available at SSRN 3908732.

Dougherty J, Kohavi R, Sahami M. (1995). Supervised and unsupervised discretization of continuous features. In *Machine learning proceedings 1995* (pp. 194–202). Morgan Kaufmann.

Emenogu, N. G., Adenomon, M. O., & Nwaze, N. O. (2019). Modeling and forecasting daily stock returns of Guaranty Trust Bank Nigeria Plc using ARMA-GARCH models, persistence, half-life volatility and backtesting. *Science World Journal, 14*(3), 1–22.

Endo, S., Cai, Z., Benjamin, S. C., & Yuan, X. (2021). Hybrid quantum-classical algorithms and quantum error mitigation. *Journal of the Physical Society of Japan, 90*(3), 032001.

Feinstein, Z. (2022). Clearing prices under margin calls and the short squeeze. *SIAM Journal on Financial Mathematics, 13*(4), SC113–SC122.

Ferreira, F. G., Gandomi, A. H., & Cardoso, R. T. (2021). Artificial intelligence applied to stock market trading: A review. *IEEE Access, 9*, 30898–30917.

Fleuret, F. (2004). Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research, 5*(9), 1531–55.

García García, F., Guijarro Martínez, F., Moya Clemente, I., & Oliver, M. J. (2012). Estimating returns and condicional volatility: A comparison between the ARMA-GARCH-M models and the backpropagation neural network. *International Journal of Complex Systems in Science, 2*(1), 21–26.

Ghani IM, Rahim HA. (2019, June). Modeling and forecasting of volatility using ARMA-GARCH: case study on Malaysia natural rubber prices. In *IOP Conference Series:* Materials *Science and Engineering* (Vol. 548, No. 1, p. 012023). IOP Publishing.

Godfrey, K. R. (2016). Detecting the great short squeeze on Volkswagen. *Pacific-Basin Finance Journal, 40*, 323–334.

Grachev OY. (2017). Application of time series models (ARIMA, GARCH, and ARMA-GARCH) for stock market forecasting.

Grover, L. K. (2005). Fixed-point quantum search. *Physical Review Letters, 95*(15), 150501.

Guerreschi, G. G. (2019). Repeat-until-success circuits with fixed-point oblivious amplitude amplification. *Physical Review A, 99*(2), 022306.

Han, H., Teng, J., Xia, J., Wang, Y., et al. (2021). Predict high-frequency trading marker via manifold learning. *Knowledge-Based Systems, 213*, 106662.

Hansen, K. B. (2020). The virtue of simplicity: On machine learning models in algorithmic trading. *Big Data & Society, 7*(1), 2053951720926558.

Haq EU, Braud T, Lee LH, Vallapuram AK et al (2022). Short, Colorful, and Irreverent! A Comparative Analysis of New Users on WallstreetBets During the Gamestop Short-squeeze. InCompanion Proceedings of the Web Conference 2022 2022 Apr 25 (pp. 52-61)

Henneke, J. S., Rachev, S. T., Fabozzi, F. J., & Nikolov, M. (2011). MCMC-based estimation of Markov Switching ARMA–GARCH models. *Applied Economics, 43*(3), 259–271.

Herman D, Googin C, Liu X, Galda A, Safro I et al (2022). A survey of quantum computing for finance. https://arxiv.org/abs/2201.02773.

Hilliard, J. E., & Hilliard, J. (2023). The GameStop short squeeze: Put–call parity and the effect of frictions before, during and after the squeeze. *Journal of Futures Markets, 43*(5), 635–661.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(8), 832–844.

Hu Y, Tao Z, Xing D, Pan Z, Zhao J, Chen X. (2020, August). Research on stock returns forecast of the four major banks based on ARMA and GARCH model. In *Journal of* Physics*: Conference Series* (Vol. 1616, No. 1, p. 012075). IOP Publishing.

Huang CW, Narayanan SS. (2017, July). Deep convolutional recurrent neural network with attention mechanism for robust speech emotion recognition. In 2017 IEEE international conference on multimedia and expo (ICME) (pp. 583–588). IEEE.

Hussain, W., Gao, H., Raza, M., Rahbi, F. A., & Merigó, J. M. (2022). Assessing cloud QoS predictions using OWA in neural network methods. *Neural Computing and Applications, 34*, 14895–14912.

Jacob, F. (2015). *Risk estimation on high frequency financial data: Empirical analysis of the DAX 30*. Springer.

Jarrow R, Li S. (2021). Media Trading Groups and Short Selling Manipulation. Available at SSRN 3804130.

Jiang Z, Liu B, Schrowang A, Xu W. (2021). Short squeezes. Available at SSRN 2019361.

Khashei, M., & Bijari, M. (2012). A new class of hybrid models for time series forecasting. *Expert Systems with Applications, 39*(4), 4344–4357.

Kim, K., Lee, S. Y. T., & Kauffman, R. J. (2023). Social informedness and investor sentiment in the GameStop short squeeze. *Electronic Markets, 33*(1), 23.

Li, X., & Zhou, J. (2022). An adaptive hybrid fractal model for short-term load forecasting in power systems. *Electric Power Systems Research, 207*, 107858.

Ling, S., & McAleer, M. (2003). Asymptotic theory for a vector ARMA-GARCH model. *Econometric Theory, 19*(2), 280–310.

Liu D, Zhang L. (2010). China stock market regimes prediction with artificial neural network and markov regime switching. In *Proceedings of the world congress on engineering* (Vol. 1, pp. 378–383).

Liu, H., & Long, Z. (2020). An improved deep learning model for predicting stock market price time series. *Digital Signal Processing, 102*, 102741.

Long, C, Lucey BM, Yarovaya L. (2021). I Just Like the Stock'versus' Fear and Loathing on Main Street: The Role of Reddit Sentiment in the GameStop Short Squeeze

Ma M, Mao Z. (2019). Deep recurrent convolutional neural network for remaining useful life prediction. In *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)* (pp. 1–4). IEEE.

Mahajan RP. (2011, February). Hybrid quantum inspired neural model for commodity price prediction. In *13th International Conference on Advanced Communication Technology (ICACT2011)* (pp. 1353–1357). IEEE.

Mancini, A., Desiderio, A., Di Clemente, R., & Cimini, G. (2022). Self-induced consensus of Reddit users to characterise the GameStop short squeeze. *Scientific Reports, 12*(1), 1–11.

Meyer, P. E., Schretter, C., & Bontempi, G. (2008). Information-theoretic feature selection in microarray data using variable complementarity. *IEEE Journal of Selected Topics in Signal Processing, 2*(3), 261–274.

Naseem, S., Fu, G. L., Mohsin, M., Zia-ur-Rehman, M., & Baig, S. (2018). Volatility of pakistan stock market: A comparison of Garch type models with fve distribution. *Amazonia Investiga, 7*(17), 486–504.

Nielsen, M. A., & Chuang, I. L. (2001). Quantum computation and quantum information. *Physics Today, 54*(2), 60.

Olteanu M, Rynkiewicz J, Maillet B. (2004). Non-linear analysis of shocks when financial markets are subject to changes in regime. In *European Symposium on Artificial Neural Networks* (pp. 87–92).

Ou, J., Li, W., & Huang, J. (2023). Frequency-domain enhanced bi-directional recurrent quantum network for stock price trend prediction. *Multimedia Tools and Applications, 83*(18), 53837–73.

Paliewicz, N. S. (2023). Playing Robinhood: Jamming Wall Street with Dumb Money in the Great Short Squeeze. *Communication Studies, 74*(3), 251–267.

Paquet, E., & Soleymani, F. (2022). QuantumLeap: Hybrid quantum neural network for financial predictions. *Expert Systems with Applications, 195*, 116583.

Qasim, R., Bangyal, W. H., Alqarni, M. A., & Ali, Almazroi A. (2022). A fine-tuned BERT-based transfer learning approach for text classification. *Journal of Healthcare Engineering, 2022*(1), 3498123.

Quinlan, J. R. (1993). *C4, 5: Programs for Machine Learning, Morgan Kaufmann* (p. 1993). PublishersInc.

Raubitzek, S., & Neubauer, T. (2021). A fractal interpolation approach to improve neural network predictions for difficult time series data. *Expert Systems with Applications, 169*, 114474.

Roberto, G. F., Lumini, A., Neves, L. A., et al. (2021). Fractal neural network: A new ensemble of fractal geometry and convolutional neural networks for the classification of histology images. *Expert Systems with Applications, 166*, 114103.

Salamat, S., Lixia, N., Naseem, S., Mohsin, M., Zia-ur-Rehman, M., & Baig, S. A. (2020). Modeling cryptocurrencies volatility using GARCH models: A comparison based on normal and student's T-Error distribution. *Entrepreneurship and Sustain, 7*(3), 1580–1596.

Santos AAP, dos Santos Coelho L, Klein CE. (2010, July). Forecasting electricity prices using a RBF neural network with GARCH errors. In *The 2010 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). IEEE.

Dos Santos Gonçalves CP. (2019). Quantum neural machine learning: Theory and experiments. *Machine Learning in Medicine and Biology*, 95–115

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science, 2*(3), 160.

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review, 5*(1), 3–55.

Sokolovsky A, Arnaboldi L. (2020). Machine Learning Classification of Price Extrema Based on Market Microstructure and Price Action Features. A Case Study of S&P500 E-mini Futures.

Solibakke, P. B. (2001). Efficiently ARMA–GARCH estimated trading volume characteristics in thinly traded markets. *Applied Financial Economics, 11*(5), 539–556.

Sun, J., Xiao, K., Liu, C., Zhou, W., & Xiong, H. (2019). Exploiting intra-day patterns for market shock prediction: A machine learning approach. *Expert Systems with Applications, 127*, 272–281.

Syuhada, K., Tjahjono, V., & Hakim, A. (2023). Improving Value-at-Risk forecast using GA-ARMA-GARCH and AI-KDE models. *Applied Soft Computing, 148*, 110885.

Tacchino, F., Macchiavello, C., Gerace, D., & Bajoni, D. (2019). An artificial neuron implemented on an actual quantum processor. *Npj Quantum Information, 5*(1), 1–8.

Tseng, F. M., Yu, H. C., & Tzeng, G. H. (2002). Combining neural network model with seasonal time series ARIMA model. *Technological Forecasting and Social Change, 69*(1), 71–87.

Umar, Z., Gubareva, M., Yousaf, I., & Ali, S. (2021). A tale of company fundamentals vs sentiment driven pricing: The case of GameStop. *Journal of Behavioral and Experimental Finance, 30*, 100501.

Vasileiou, E. (2021). Does the short squeeze lead to market abnormality and antileverage effect? Evidence from the Gamestop case. *Journal of Economic Studies, 49*(8), 1360–73.

Vasileiou E, Bartzou E, Tzanakis P. (2021). Explaining Gamestop Short Squeeze using Intraday Data and Google Searches. Available at SSRN 3805630.

Vaughan, M., Gruber, J. B., & Langer, A. I. (2023). The tension between connective action and platformisation: Disconnected action in the GameStop short squeeze. *New Media & Society, 27*(2), 632–54.

Wan, K. H., Dahlsten, O., Kristjánsson, H., Gardner, R., & Kim, M. S. (2017). Quantum generalisation of feedforward neural networks. *Npj Quantum Information, 3*(1), 1–8.

Wang, C., Yang, Y., Xu, L., & Wong, A. (2023). A Hybrid Model of Primary Ensemble Empirical Mode Decomposition and Quantum Neural Network in Financial Time Series Prediction. *Fluctuation and Noise Letters, 22*(04), 2340006.

Wu, J., Xu, K., Chen, X., Li, S., & Zhao, J. (2022). Price graphs: Utilizing the structural information of financial time series for stock prediction. *Information Sciences, 588*, 405–424.

Xiang NW, Dabbagh M. (2022, September). Stock Market Price Prediction: Text Analytics of the GameStop Short Squeeze. In *2022 IEEE International* Conference *on Artificial Intelligence in Engineering and Technology (IICAIET)* (pp. 1–6). IEEE.

Zhang, G., Ali, S., Wang, X., Wang, G., Pan, Z., & Zhang, J. (2019). SPI-based drought simulation and prediction using ARMA-GARCH model. *Applied Mathematics and Computation, 355*, 96–107.

Zheng, X., Tian, H., Wan, Z., Wang, X., et al. (2021). Game starts at GameStop: Characterizing the collective behaviors and social dynamics in the short squeeze episode. *IEEE Transactions on Computational Social Systems, 9*(1), 45–58.

Zidan, M., Abdel-Aty, A. H., El-shafei, M., Feraig, M., et al. (2019). Quantum Classification Algorithm Based on Competitive Learning Neural Network and Entanglement Measure. *Applied Sciences, 9*, 1277.