Information Fusion 113 (2025) 102616

Contents lists available at ScienceDirect

Information Fusion

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

Full length article

Data-driven stock forecasting models based on neural networks: A review

Wuzhida Bao [a], Yuting Cao [b], Yin Yang [b], Hangjun Che [c], Junjian Huang [c], Shiping Wen [a,*]

[a] *Australian AI Institute, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney 2007, Australia*
[b] *College of Science and Engineering, Hamad Bin Khalifa University, Doha 5855, Qatar*
[c] *Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China*

ARTICLE INFO

Keywords:
Stock forecast
Finance
Financial market
Neural network
Deep learning

ABSTRACT

As a core branch of financial forecasting, stock forecasting plays a crucial role for financial analysts, investors, and policymakers in managing risks and optimizing investment strategies, significantly enhancing the efficiency and effectiveness of economic decision-making. With the rapid development of information technology and computer science, data-driven neural network technologies have increasingly become the mainstream method for stock forecasting. Although recent review studies have provided a basic introduction to deep learning methods, they still lack detailed discussion on network architecture design and innovative details. Additionally, the latest research on emerging large language models and neural network structures has yet to be included in existing review literature. In light of this, this paper comprehensively reviews the literature on data-driven neural networks in the field of stock forecasting from 2015 to 2023, discussing various classic and innovative neural network structures, including Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Transformers, Graph Neural Networks (GNNs), Generative Adversarial Networks (GANs), and Large Language Models (LLMs). It analyzes the application and achievements of these models in stock market forecasting. Moreover, the article also outlines the commonly used datasets and various evaluation metrics in the field of stock forecasting, further exploring unresolved issues and potential future research directions, aiming to provide clear guidance and reference for researchers in stock forecasting.

1. Introduction

With the advancement of society and economic growth, financial markets have increasingly played a pivotal role as catalysts for financial development and wealth creation [1]. Advances in financial research have not only deepened economic theories but also fostered innovation in financial instruments and markets. The stock market, as a crucial component of the financial market, has always been a critical indicator of economic health and developmental trends, serving as an essential platform for companies to raise capital and for investors to participate in corporate growth and prosperity.

The predictability of the stock market has always been a hot topic of interest in both the financial and academic communities. As early as the beginning of the last century, scholars studied the behaviour of the stock market, with Louis Bachelier first proposing in 1900 that stock price fluctuations follow a random walk process [2]. Subsequently, scholars like Cootner and Fama [3,4] thoroughly explored and tested this theory, supporting the view that short-term changes exhibit random walk characteristics, suggesting that stock market prices form a random drift and thus are unpredictable. Building on the random walk theory, Professor Fama from the University of Chicago introduced the renowned Efficient Market Hypothesis (EMH) in the 1960s [5]. EMH considers the market "information efficient", which implies that market prices fully and instantaneously reflect all available information, encompassing both public and insider information. This notion suggests that market prices are inherently unpredictable, as any new information or changes are immediately incorporated into the market prices, rendering it challenging for investors to profit from the market by analysing publicly available information alone. The Efficient Market Hypothesis typically comes in three forms: Weak form: stock prices reflect all information contained in historical price sequences, meaning technical analysis cannot be used to predict future stock price movements. Semi-strong form: stock prices reflect all publicly available information, including historical prices and other published data. Under this form, even fundamental analysis cannot yield abnormal

* Corresponding author.
E-mail addresses: wuzhida.bao@student.uts.edu.au (W. Bao), ycao@hbku.edu.qa (Y. Cao), yang@hbku.edu.qa (Y. Yang), hjche@swu.edu.cn (H. Che), hmomu@sina.com (J. Huang), shiping.wen@uts.edu.au (S. Wen).

https://doi.org/10.1016/j.inffus.2024.102616
Received 27 May 2024; Received in revised form 31 July 2024; Accepted 1 August 2024
Available online 5 August 2024

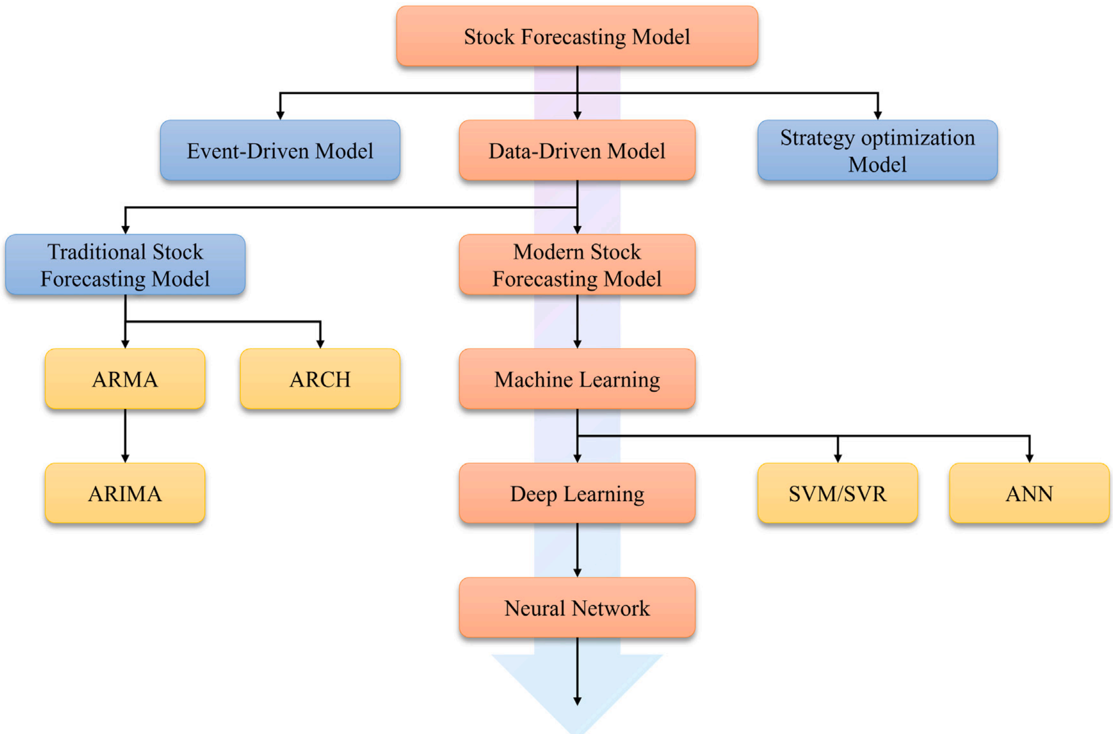

Fig. 1. The framework of stock forecasting research. Different colours represent different meanings. Red indicates the research direction of this paper, blue signifies other important branches, and yellow denotes specific methods.

returns. Strong form: stock prices reflect all public and private information, meaning even insider trading cannot consistently outperform the market.

Despite the EMH and RWT laying an important theoretical foundation for stock market prediction research, the limitations of these traditional theories have been increasingly questioned over time as financial markets become more complex and data technology advances rapidly [6]. The desire of market participants to surpass average market returns has spurred new forecasting methods and trading strategies. Consequently, developing strategies that can accurately predict stock prices and exceed average market returns has become a focal point for financial analysts, investors, and policymakers.

In the field of stock market prediction, research can be categorized into three approaches based on trading strategies: data-driven [7], event-driven [8], and strategy optimization [9]. Data-driven methods emphasize using information contained in historical data to predict future stock price movements, typically relying on complex mathematical models and algorithms such as time series analysis, machine learning, and deep learning to capture and learn the underlying patterns of stock price fluctuations. Event-driven methods focus on the impact of specific events on stock prices, such as mergers and acquisitions, earnings releases, and policy changes, arguing that market fluctuations are influenced not only by historical data but also significantly by sudden events. Strategy optimization builds upon data-driven and event-driven foundations, further considering how to construct and optimize investment strategies to achieve the best risk-adjusted returns. Given the continuous changes in the stock market and the real-time updating nature of data, data-driven forecasting methods are critical due to their advantage of directly utilizing real-time updated data. Unlike event-driven and strategy optimization methods, data-driven approaches are more capable of intuitively acquiring and processing vast amounts of market data, effectively mining and analysing future trends in the stock market.

As early representatives of data-driven prediction strategies, statistical methods have provided a theoretical and practical foundation. Limited by computational power, initial stock market prediction methods are linear regression [10] and time series decomposition [11]. However, these simple statistical models often struggle to capture the complexity and nonlinear characteristics of stock price movements, resulting in lower prediction accuracy. Researchers introduced complex time series models to overcome these flaws, including the classic autoregressive moving average (ARMA) and autoregressive conditional heteroskedasticity (ARCH) models. The ARMA model combines autoregression (AR) [12] and moving average (MA) [13] mechanisms, capturing the linear dependencies in stock price sequences, while the ARCH model introduces the concept of conditional heteroskedasticity, which describes the clustering effect of stock price volatility, where the magnitude of stock price fluctuations depends on previous fluctuations. Although these improved methods have made some progress, they still have shortcomings. To further enhance the modelling capacity for complex stock price sequences, researchers proposed the ARIMA (autoregressive integrated moving average) model [14]. The ARIMA model adds an integration mechanism to the ARMA model to better eliminate non-stationarity and stabilize the series. The (AR) component describes the linear relationship between stock prices and their past values, the (I) component eliminates non-stationarity through differencing, making the series stable, and the (MA) component models the autocorrelation of series residuals. By appropriately setting these three components' parameters appropriately, the ARIMA model can more flexibly fit complex stock price sequences. Kumar et al. [15] proposed a hybrid model combining ARIMA with support vector machines (SVM), artificial neural networks (ANN), and random forests (RF), respectively, which leverages ARIMA's advantages in capturing linear patterns and the machine learning models' capabilities in fitting nonlinear patterns. Research has found that the ARIMA-SVM hybrid model achieves the best prediction accuracy and investment returns. Similarly, Wang et al. [16] combined XGBoost with ARIMA to propose a DWT-ARIMA-GSXGB hybrid model, which uses discrete wavelet transform (DWT) to decompose the original stock price data into approximate and residual parts. Then, it predicts these parts using the ARIMA model and an enhanced XGBoost (GSXGB) model before combining the prediction results through wavelet reconstruction. This hybrid modelling approach leverages the strengths of ARIMA in capturing linear patterns and XGBoost in fitting nonlinear patterns, significantly improving prediction performance and surpassing traditional ARIMA and other statistical methods. Overall, the ARIMA model, as a classic statistical method, remains a crucial benchmark in financial time series forecasting research. It provides a solid theoretical and practical foundation for complex prediction models based on machine learning due to its excellent ability to capture linear patterns.

Despite significant advances in stock market forecasting based on statistical models, the increasing complexity of financial markets has led to new data-driven prediction technologies. Substantial improvements in computational capabilities have established modern forecasting methods represented by machine learning and deep learning as mainstream in stock market prediction. Among these, artificial neural networks (ANN) [17,18] and support vector machines (SVM) [19] have shown great potential in the field of stock market forecasting. ANNs, machine learning models that mimic biological neural networks, consist of input, hidden, and output layers capable of end-to-end learning of complex feature representations from input data, mapping them to price predictions. As an excellent supervised learning algorithm, SVM achieves precise classification or regression by finding the optimal hyperplane to separate different categories of samples. Unlike ANNs, SVMs are insensitive to the amount and dimension of samples, showing superior performance in handling high-dimensional financial data. Wang et al. [20] proposed a hybrid model that combines ANNs with ARIMA to leverage the advantages of modelling both linear and nonlinear behaviours. This model achieved good prediction results on multiple datasets, including Wolf sunspot data, Canadian lynx data, and IBM stock prices. Combining stock box theory and SVM, Wen et al. [21] introduced a new type of intelligent trading system based on oscillation box predictions. This system uses two SVM estimators to predict the upper and lower limits of price oscillations, then constructs a trading strategy based on these constrained predictions, achieving leading results on the S&P 500 dataset. At the same time, other machine learning methods such as k-nearest neighbours (KNN) [22], random forests (RF) [23], and decision trees (DT) [24] are also frequently applied in stock market forecasting. Compared to traditional statistical models, machine learning demonstrates a stronger nonlinear modelling capability, enabling it to capture complex dynamic patterns in stock price time series effectively. However, machine learning methods still have limitations in handling large-scale data, automatic feature extraction, and model generalization. They often rely on carefully designed features or manually selected model parameters, which increases the complexity of model development and limits the models' adaptability and flexibility under unknown or changing market conditions.

Although traditional statistical and machine learning methods have made significant progress in stock market forecasting, the tremendous leaps in computational power and the substantial growth in data volumes have propelled neural network-based deep learning methods to the forefront of the field. These methods are well-suited for handling large and high-dimensional financial datasets, ushering in a new era of data-driven prediction methodologies. A diverse array of model structures, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), gated recurrent units (GRUs), graph neural networks (GNNs), generative adversarial networks (GANs), and large language models (LLMs), offer various perspectives for Stock Forecasting, demonstrating the extensive application and significant potential of deep learning in financial market analysis. Considering that earlier review studies did not fully cover these advanced technologies, this research introduces a series of new perspectives, concentrating on the application and recent advancements of data-driven neural network models in stock forecasting. Specifically, this paper contributes in the following three areas:

- **Comprehensive Technology Review:** This paper thoroughly examines the application of data-driven neural networks in stock forecasting from 2015 to 2023, with a particular focus on the latest technological advancements, including Transformers, generative adversarial networks (GANs), and large language models (LLMs), addressing gaps in recent technology and article coverage in the existing literature.
- **In-depth Methodological Analysis:** The paper focuses on data-driven neural network models, deeply assessing the performance of various neural network structures in stock price prediction, including their structural composition, innovations, and advantages, and provides detailed insights into their performance in practical forecasting.
- **Current Challenges and Future Directions:** The paper not only highlights the critical issues faced by neural network models in the field of stock market forecasting but also explores potential solutions and future research directions, offering insights and inspiration for the development of this field.

The rest of the paper is structured as follows: Section 2 reviews previous studies on this topic, discusses their achievements and limitations, and elucidates the contributions of this paper. Section 3 provides a detailed introduction to various types of data-driven neural network models for stock forecasting, classifying them based on their core network structures while discussing their innovative features and achievements. Section 4 introduces the stock forecasting field's commonly used datasets and evaluation metrics. Section 5 summarizes the significant issues and deficiencies currently present in this research area and proposes future research directions. Finally, Section 6 concludes the paper. Fig. 1 displays the overall research framework of existing stock market forecasting models.

2. Related work

As financial markets become increasingly complex and changeable, traditional statistical and econometric models have gradually revealed their limitations in capturing the nonlinear and dynamic characteristics of stock prices. With the rapid development of information technology and the surge in network data volume, research focus has shifted towards machine learning and deep learning methods, particularly through in-depth studies in the field of neural networks. These methods have driven the development of various innovative network structures, which are widely used to predict dynamic stock price changes accurately. In their research, Atsalakis et al. [25] reviewed over 100 papers utilizing neural networks and neuro-fuzzy techniques for stock price prediction, meticulously categorizing them based on input data types, prediction techniques, performance assessments, and application performance metrics. Li and Ma [26] discussed methods for predicting stock and option prices using nonlinear artificial neural network models and reviewed the applications of artificial neural networks in banking operations and financial crisis prediction. Soni et al. [27] examined the effectiveness of artificial neural networks (ANN) in stock market forecasting and found that ANNs demonstrated more substantial competitiveness and a key role compared to genetic algorithms and multiple linear regression methods. Tkáč and his team [28] analysed 412 papers from 1994 to 2015, focusing on financial distress, bankruptcy identification, stock price forecasting, and decision support systems, particularly in classification tasks. Dattatray P. Gandhmal and K. Kumar [29] systematically reviewed 50 papers on stock forecasting from 2010 to 2018, discussing them based on the forecasting techniques used, publication years, evaluation metrics, datasets, and tools. These studies encompass a range from traditional Bayesian models and fuzzy classifiers to modern machine learning methods such as artificial neural networks (ANN) and support vector machines (SVM).

Moreover, these reviews highlighted existing research gaps and challenges, providing suggestions for future research directions in the field of stock market forecasting. Some studies also focus on event-driven stock investment strategies. For example, Nassirtoussi et al. [30] combined data-driven and event-driven concepts, proposing that public sentiment analysis in social media and online news has a direct impact on financial market predictions, potentially leading to significant profits or losses. Based on this concept, they reviewed the application of online text mining in market forecasting, compared different systems, and identified their fundamental differences. Li et al. [31] systematically reviewed 229 research articles from 2007 to 2016 from the fields of finance, management information systems, and computer science, discussing the interaction between online media and the stock market, classifying state-of-the-art research, and summarizing the main techniques for transforming textual information into machine-processable formats. Xing et al. [32] conducted an in-depth investigation of the natural language processing-based financial forecasting (NLFF) field, systematically categorizing and analysing the technological applications in this area, clarifying the boundaries of NLFF research. This work aims to enhance understanding of the progress and hot issues in NLFF, promoting interdisciplinary, in-depth communication. Similarly, Shahi et al. [33] conducted a normalized comparison of the stock market prediction performance of LSTM and GRU under the same conditions, objectively assessing the value of financial news sentiment analysis in stock price prediction.

In recent years, deep learning technology has become the mainstream method for stock market forecasting. Sezer et al. [34] conducted a thorough discussion of the application of deep learning in financial forecasting from 2005 to 2019, categorizing the studies by application areas (such as indices, foreign exchange, commodities forecasting) and deep learning models (such as CNN, DBN, LSTM). After reviewing 124 papers over the past three years, Weiwei Jiang [35] systematically categorized data sources, neural network architectures, and evaluation metrics, focusing on the implementability and reproducibility of the studies. In Eckerli's research [36], the author tested three known GAN structures on financial time series data to assess their performance in financial modelling. Although no specific expected outcomes were set before the experiments, the results demonstrated the significant effectiveness of GANs in generating financial time series. By applying different GAN architectures to real financial data and successfully evaluating the statistical properties of the generated data, the practicality and potential value of GANs in the financial sector were validated. Kumbure et al. [37] reviewed 138 journal articles from 2000 to 2019. They focused on dataset features, machine learning techniques, and their derivative methods, categorizing the research methods into supervised and unsupervised machine learning. Zou et al. [38] provided a structured review of deep learning methods for stock market forecasting, analysed 94 papers, and subdivided the forecasting tasks into subdomains such as stock trend, stock price forecasting, portfolio management, and trading strategies while also discussing unresolved issues and looking forward to future research directions. Masini et al. [39] studied the latest advancements in supervised machine learning and time series forecasting for high-dimensional models, differentiating methods into linear and nonlinear models and highlighting the importance of nonlinear models combined with large datasets for economic forecasting.

Although previous reviews covered everything from traditional statistical econometric methods to older machine learning techniques, they have gradually moved away from current research hotspots in the field. Although some recent review studies provide a basic overview of currently popular deep learning methods, they still need to detail specific network structure designs and parameter details. At the same time, with the rise of large language models and ongoing research in neural network structures, some of the latest methods have not been covered in these reviews. Given the shortcomings of existing review studies, this research brings a new perspective and focus, comprehensively tracking and summarizing data-driven neural network methods for stock forecasting. As previously stated, the aim and contribution of this research are to address the gaps in existing literature, providing a more comprehensive and in-depth review of this field.

3. Neural network for stock forecasting

Neural networks are powerful computational models that mimic the human brain's information processing capabilities, accomplishing complex pattern recognition and learning through extensive networks of neurons. Without explicit programming, these models can autonomously extract useful representations and features from massive datasets. This capability has made neural networks indispensable tools in various fields, including image processing, voice recognition, natural language processing, and complex predictive tasks. In the financial sector, neural networks such as stock prices, currency exchange rates, and market trends are extensively used for forecasting the stock market. Our survey encompasses various classical and novel neural network architectures, including RNNs, CNNs, Transformers, GNNs, GANs, and LLMs, exploring their applications in stock market forecasting. The neural network stock forecasting methodology reviewed in this paper is shown in Fig. 2.

3.1. Stock forecasting models based on Recurrent Neural Network

Recurrent Neural Networks (RNNs) [56] are specifically designed for processing sequential data and play a crucial role in stock market forecasting. Unlike traditional feedforward neural networks, RNNs can effectively capture temporal dependencies within sequence data by utilizing information from past hidden states to process current inputs. Specifically, RNNs iteratively process stock price inputs at each time step, updating hidden states influenced by both the current input and previous states. Additionally, the unique features of RNNs, such as variable-length input handling and parameter sharing, significantly enhance their applicability and computational efficiency in stock market forecasting. Table 1 provides the specific details of the reviewed RNN methods.

Rather et al. [40] combined RNNs with Exponential Smoothing (ES) and Autoregressive Moving Average (ARMA) models to develop a novel and robust Hybrid Prediction Model (HPM). This model excels at predicting abrupt changes and spikes (nonlinear patterns) in data, effectively enhancing predictive performance and significantly reducing forecast errors. An experiment on India's National Stock Exchange (NSE) demonstrated that the HPM outperforms the vanilla RNN, achieving higher prediction accuracy. Berradi Zahra and Lazaar Mohamed [41] integrated Principal Component Analysis (PCA) with RNNs, achieving satisfactory forecasting results on the Casablanca Stock Exchange data. This approach reduces the dimensionality of the features input to the RNN using PCA, allowing the RNN to focus more on the most relevant features, thus enhancing the model's predictive performance.

However, vanilla RNNs face a severe "vanishing gradient" problem, where gradients diminish to nearly zero as they propagate through longer sequences. That leads to decreased performance when learning from longer temporal data and difficulty capturing long-term dependencies. To overcome this flaw, Hochreiter et al. introduced the Long Short-Term Memory (LSTM) [57]. LSTM is an enhanced RNN model that incorporates cell states and three gating mechanisms — forget, input, and output gates — to regulate the flow of information. These gating mechanisms allow LSTMs to retain or forget information effectively, making them more stable and efficient when processing long sequence data. A standard LSTM cell state equation is as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$g_t = \tanh(W_g \cdot [h_{t-1}, x_t] + b_g) \tag{3}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{4}$$

$$c_t = f_t * c_{(t-1)} + i_t * g_t \tag{5}$$

$$h_t = o_t * \tanh(c_t) \tag{6}$$

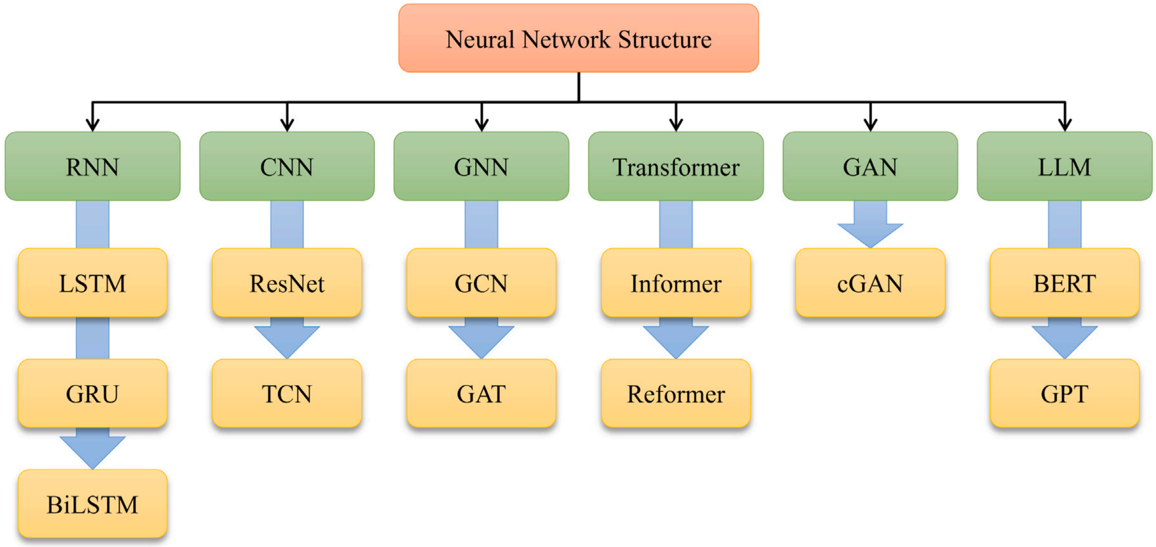

Fig. 2. The neural network structure and its well-known variants reviewed in this paper.

Table 1
Stock forecasting models details based on recurrent neural network.

Author	Model	Method	Year	Research object	Evaluate metrics	Dataset
Rather [40]	Hybrid-RNN	RNN	2015	Price prediction	MSE, MAE	BSE
Berradi [41]	RNN	RNN, PCA	2019	Price prediction	MSE	CSE
Liu [42]	LSTM	LSTM	2017	Price prediction	MSE, MSPE	–
Liu [43]	LSTM	LSTM	2018	Trend prediction	Accuracy	CSI300
Abdul [44]	MLS LSTM	LSTM	2023	Price prediction	R2, Adjusted R2, RMSE, MAPE	Samsung
Baek [45]	ModAugNet	LSTM	2018	Price prediction	MSE, MAE, MAPE	S&P500, KOSPI200
Kim [46]	Hybrid-LSTM	LSTM, CARCH, DFN	2018	Price prediction	MSE, MAE, HMSE, HMAE	KOSPI200
Kumar [47]	RNN-LSTM	LSTM, RNN	2021	Price prediction	Error rate, Precision, Recall, F1	LON, NSE, NYSE, BOM, NASDAQ
Yang [48]	LSTM-EEMD	LSTM, EMD	2020	Price prediction	R2, MAE, RMSE	ASX, DAX, HSI, S&P500
Zhang [49]	CEEMD-PCA-LSTM	LSTM, CEEMD, PCA	2020	Price prediction	MAE, RMSE, NMSE, DS	DJI, S&P500
Lin [50]	CEEMDAN-LSTM	LSTM, CEEMDAN	2022	Price prediction	MSE, MAE, HMSE, HMAE, MCS	CSI300, S&P500, STOXX50
Yang [51]	BiLSTM	BiLSTM	2022	Price prediction	R2, RMSE MAPE,	CSI300
Vaziri [52]	PSO-BiLSTM	BiLSTM, PSO, MOMP	2023	Price prediction	R2, MSE, RMSE, MAPE	TSE, OTC
Minh [53]	TGRU	GRU, Stock2Vec	2018	Trend prediction	Accuracy, Precision, Recall	S&P500
Li [54]	ST-GRU	GRU, CMSCE	2020	Price prediction	MAE, RMSE, SMAPE, TIC, DS, CP, CD	WTI, BRE, NG, HO
Gupta [55]	StockNet	GRU	2022	Price prediction	MAE, RMSE, MAPE	NIFTY50

In this table, **Price prediction** seeks to precisely forecast the future trading price of a stock at a specified time. **Trend prediction** aims to identify and predict the general direction of stock price movements, offering a broad perspective on market trends to aid in investment decision-making.

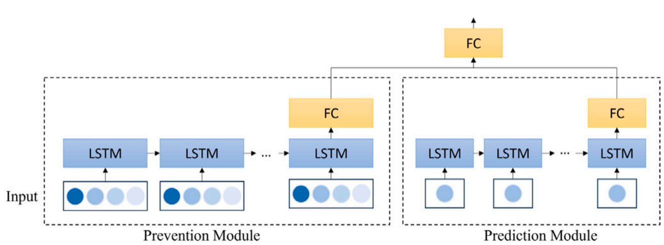

Fig. 3. ModAugNet architecture.

where σ represents the sigmoid function, which outputs a value between 0 and 1. W denotes the weight matrix, and b represents the bias term. $[h_{t-1}, x_t]$ indicates the concatenation of the previous hidden state and the current input. f_t represents the output of the forget gate, determining which parts of the existing cell state should be discarded. i_t is the output of the input gate, identifying which new information is important and needs to be retained. g_t is the candidate cell state derived at the current timestep, providing new information for potential updates to the cell state. The current cell state c_t is updated by filtering the previous cell state $c_{(t-1)}$ through f_t and incorporating it with g_t adjusted by i_t. o_t represents the output of the output gate, which determines which parts of the cell state information will influence the calculation of the next hidden state. The new hidden state h_t is obtained by multiplying the result of the output gate o_t with the tanh value of the current cell state c_t, ensuring that the network can transmit relevant information based on the cell state. This ingeniously designed process allows the LSTM unit to effectively maintain and convey information over long sequences, thus capturing long-term dependencies.

Liu et al. [42] categorized time series data into three distinct patterns: strong periodicity, periodic rising/falling trends, and extremely long cyclical data, and compared their proposed LSTM RNN model against ARIMA and GRNN for each data type. The experimental results demonstrated that the LSTM RNN model outperformed the other methods. Stacking multiple LSTM layers allows for learning the temporal dependencies of data at different levels, enabling the upper layers to capture more complex dynamics while the lower layers process features closer to the original input. Liu et al. [43] found that within a certain range of layers, the more layers, the higher the accuracy of stock data prediction on the JoinQuant platform. Similarly, Abdul Quadir Md et al. [44] also adopted a stacking strategy and introduced an innovative stock market prediction model—Multi-Layer Sequential Long Short Term Memory (MLS LSTM). This model utilizes the Adam optimizer and stacks multiple LSTM layers, analysing the relationship between past and future values using normalized time series data divided into time steps, thereby making accurate predictions. The study results showed that the MLS LSTM achieved a prediction accuracy of 95.9% on the training set and 98.1% on the test set. This remarkable performance demonstrates the advantages of using a multi-layer LSTM structure. Although increasing the number of layers may reduce the model's generality, the leading results of the MLS LSTM across multiple datasets undoubtedly prove it to be a superior architecture.

The LSTM structure has demonstrated exceptional time series modelling capabilities across various architectures. Although the straightforward application of LSTM has already surpassed traditional machine learning and artificial neural network methods, integrating LSTM with other strategies or architectures often results in better prediction outcomes. To address the overfitting problem caused by limited training data in stock price prediction, Yujin Baek and Ha Young Kim designed a framework named ModAugNet [45]. As shown in Fig. 3, based on LSTM, the framework is divided into two main modules: a Prevention LSTM module to prevent overfitting and a Prediction LSTM module for forecasting. Each module produces an output, and the final prediction is obtained by merging these outputs. To address the issue of insufficient raw data, the authors designed a data augmentation strategy by selecting five companies from a specific combination of ten companies' stock price data, generating 252 different input data combinations (C(10, 5)). These augmented data are fed into the Prevention LSTM module for learning to prevent the model from overfitting. This dual-module design and data augmentation strategy effectively enhance the model's ability to learn potential information in time series data and resolve the overfitting problem with limited data. Experimental results confirm the effectiveness of this method on the augmented dataset. Kim et al. [46] explored integrating the LSTM model with various Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models, aiming to leverage LSTM's capabilities in nonlinear time series modelling alongside GARCH's advantages in modelling time series volatility. Experiments on the KOSPI 200 demonstrated that their proposed GEW-LSTM model, which combines three different GARCH models (GARCH(1,1), EGARCH, and EWMA), achieved optimal forecasting results. This achievement confirms that combining neural network models with multiple econometric models can significantly enhance predictive performance. Although the design of the model's structure is a crucial factor in its effectiveness, optimizing model parameters also needs attention. Kumar et al. [47] argue that data volatility, time lags, and the extensive architectural parameters of neural networks are issues preventing optimal network performance. Thus, they proposed an RNN-LSTM combined with flower pollination and particle swarm optimization algorithms. This systematic approach helps to automatically generate optimized networks and, by adjusting hyperparameters, achieves more accurate learning processes, reduces errors, and improves accuracy. Comparative results on six stock datasets show that the metaheuristic-optimized RNN-LSTM network significantly enhances prediction accuracy.

The Empirical Mode Decomposition (EMD) [58] is an analysis method used to handle nonlinear and non-stationary time series data, which can decompose complex time series into multiple Intrinsic Mode Functions (IMFs). Combining EMD with LSTM networks enables the utilization of multiscale features extracted by EMD to enhance LSTM's modelling capability for complex financial time series. Yang et al. [48] utilized integrated EMD to decompose complex original stock price time series into smoother, more regular, and stable subsequences than the original time series, followed by using the LSTM method to train and predict each subsequence. Zhang et al. [49] proposed a novel deep learning model CEEMD-PCA-LSTM. This model employs Complementary Ensemble Empirical Mode Decomposition (CEEMD) for sequence smoothing and decomposition, followed by Principal Component Analysis (PCA) to reduce the dimensionality of IMF components obtained from CEEMD decomposition. This step effectively eliminates redundant information and enhances the model's prediction response speed. The refined high-level features are sequentially fed into the LSTM network to predict the next trading day's closing price for each component, and finally, these prediction results are combined to generate accurate overall stock price predictions. Lin Yu et al. [50] adopted a more complex hybrid model combining Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) and LSTM to predict the volatility of the CSI300, S&P500, and STOXX50 indices. Unlike CEEMD, which requires manual specification of noise intensity, CEEMDAN dynamically adjusts the noise magnitude according to the characteristics of the data itself, thereby obtaining better decomposition results.

Bidirectional Long Short-Term Memory (BiLSTM) [59] is an extension of LSTM that utilizes the bidirectional nature of time series data, considering both the historical information of previous time steps and the future information of subsequent time steps. That enables BiLSTM to better capture the contextual semantic relationships in the input sequence and extract richer feature representations. Building upon these advantages, Yang Mo and Wang Jing [51] compared BiLSTM with unidirectional LSTM, Support Vector Regression (SVR), and Autoregressive Integrated Moving Average (ARIMA) models, showing that the BiLSTM model can adequately capture both past and future data information, considering the reverse relationship of the data, and possesses the highest prediction accuracy. Vaziri et al. [52] utilized the particle swarm optimization algorithm to optimize BiLSTM, and the adjusted PSO-BiLSTM method outperformed traditional methods and exhibited more vital generalization ability.

The LSTM structure contains multiple gates, which will bring about a large number of parameters, making its training and tuning process quite complex and time-consuming, which proves inadequate when dealing with large-scale data. Therefore, based on LSTM, Kyunghyun Cho et al. proposed a variant that combines accuracy and efficiency: Gated Recurrent Unit (GRU) [60]. GRU adopts a simpler gating mechanism, integrating the forget gate and the input gate into one update gate while eliminating the output gate, thereby reducing the model complexity. The formula for GRU is as follows:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{7}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{8}$$

$$g_t = \tanh(W_g \cdot [r_t * h_{t-1}, x_t] + b) \tag{9}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * g_t \tag{10}$$

Similar to LSTM, σ represents the sigmoid function used to output a value between 0 and 1, W denotes the weight matrix, and b is the bias. The symbol $[h_{t-1}, x_t]$ indicates the concatenation of the previous hidden state and the current input. z_t represents the output of the update gate, determining the retention of past information and the inclusion of new information. r_t is the reset gate, determining which past information should be forgotten. Similarly, g_t denotes the candidate cell state, while h_t is the final hidden state at time step t, regulated by z_t and combined with the previous hidden state h_{t-1} and the new candidate state g_t. Compared to LSTM, the GRU structure is more straightforward with fewer parameters, thus resulting in lower computational costs. However, the emergence of GRU has not entirely replaced LSTM; both models have their advantages and disadvantages in different tasks, thus requiring the selection of an appropriate model based on the actual application scenario [61].

Inspired by [59], Minh et al. [53] proposed a dual-stream Gated Recurrent Unit network (TGRU) along with an emotion Stock2Vec embedding model trained on stock news and sentiment lexicons. Stock2Vec is first used to generate word embeddings, which are inputted into TGRU for training. The experimental results of this research show that the method proposed in this paper exceeds the previous limitation of relying solely on shallow feature analysis, effectively capturing the deep structural relationships within financial news vocabulary, thereby providing a more refined tool for financial news analysis. Li Jingmiao et al. [54] embed the stochastic event intensity function into GRU, proposing a new ST-GRU model. The stochastic event intensity function weights historical data based on occurrence time, where newer

Table 2
Stock forecasting models details based on convolutional neural network.

Author	Model	Method	Year	Research object	Evaluate metrics	Dataset
Selvin [62]	CNN	CNN	2017	Price prediction	Error percentage	NSE
Hoseinzade [63]	CNNpred	CNN	2019	Price prediction	Macro-Averaged-F-Measure	S&P500, NASDAQ, DJI, NYSE, RUSSELL
Durairaj [64]	CNN	CNN, Chaos, PR	2022	Price prediction	MSE, MAPE, Dstat, Theil's U.	S&P500, Nifty50, SSE
Börjesson [65]	CNN	CNN, Residual	2020	Trend prediction	MAPE	S&P500
Zhao [66]	LightGBM-Denoised-ResNet	CNN, ResNet, LightGBM	2020	Price prediction	MSE, MAE, RMSE	USDJPY
Liu [67]	ResNet 50	ResNet, Bagging method	2018	Price prediction	AUC	–
Zhang [68]	Hybrid CNN-LSTM-ResNet	CNN, ResNet, LSTM	2023	Trend prediction	Accuracy, Precision, Recall, F1	DJI
Liu [69]	M-GTCN	TCN, GLU	2019	Price prediction	MSE, MAE, RMSE, MAPE	Microsoft Stock
Deng [70]	KDTCN	TCN, KGs	2019	Trend prediction	Accuracy, F1	DJIA
Yao [71]	MEMD-TCN	TCN, MEMD	2023	Price prediction	RMSE, MAPE, DA, MASE	SSEC, DJI, FTSE100, FCHI, NIKKEI225, IRTS, STI
Kanwal [72]	BiCuDNNLSTM-1dCNN	CNN, BiLSTM, Cuda	2022	Price prediction	MAE, RMSE	DAX, HSI
Wu [73]	SACLSTM	CNN, LSTM	2023	Trend prediction	Accuracy	Ten stocks
Livieris [74]	CNN-LSTM	CNN, LSTM	2020	Price prediction	MAE, RMSE, Accuracy, AUC, SEN, SPE	USD Daily gold prices
Livieris [75]	CNN-LSTM	CNN, LSTM	2021	Price prediction	MAE, RMSE, R2, Accuracy, GM, SEN, SPE	BTC, ETH, XRP

historical data holds more value for present and future information. Regression analysis on four energy futures indices, WTI, BRE, NG, and HO, shows a strong correlation between predicted and actual data. Inspired by data augmentation techniques in ModAugNet [45], Gupta [55] developed the StockNet model to address overfitting issues. This model combines two GRU units and a novel data augmentation technique, including an injection module for preventing overfitting and an enquiry module for stock index prediction. Experiments on the Indian stock market (CNX-Nifty) demonstrate that this unique network architecture significantly enhances model performance, resulting in a 65.59% reduction in test losses for RMSE, 27.30% for MAE, and 14.89% for MAPE metrics.

Recurrent Neural Networks (RNNs) are renowned for their ability to process sequential data and capture temporal dynamics. However, RNNs face challenges in effectively learning long-term dependencies in stock market data. These networks are prone to issues such as gradient vanishing or exploding, especially when dealing with long sequences of stock market data involving complex volatility and irregular trends. Additionally, stock market data's non-linearity and high noise characteristics require models to capture and predict its time-varying behaviour, which RNNs try to address but are often limited by their architecture. Therefore, for stock prediction tasks, it is crucial to optimize the architecture of RNNs further or explore new methods to enhance their learning and predictive ability for these complex data patterns.

3.2. Stock forecasting models based on Convolutional Neural Network

Convolutional Neural Networks (CNNs) [76] are a type of deep learning model primarily used in image processing [77,78]. CNNs employ various filters within their convolutional layers to extract local features, with each filter focusing on capturing specific information. Non-linear activation functions and pooling layers help to enhance the model's generalization capabilities and reduce computational complexity. This approach allows CNNs to extend beyond visual domains to handle complex structured time series data. In applying time series forecasting, these characteristics enable CNNs to identify complex cyclical patterns and trends effectively. For instance, in stock market data analysis, varying sizes of convolutional kernels can engage in multi-scale learning, capturing features across multiple time scales. Table 2 provides the specific details of the reviewed CNN methods.

Selvin et al. [62] compared linear algorithms (such as AR, MA, ARUMA) with non-linear algorithms (including ARCH, GARCH, as well as neural network models like CNN, RNN, and LSTM) on a dataset from the National Stock Exchange of India (NSE). The study found that,

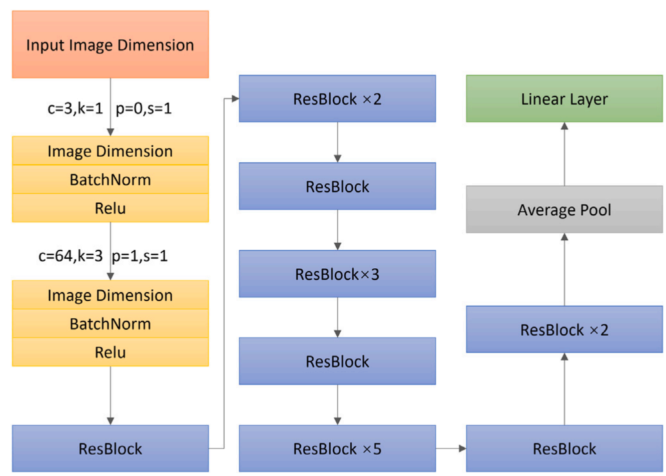

Fig. 4. Wavelet Denoised-ResNet method architecture.

although RNNs and LSTMs are designed to capture the dynamic characteristics of time series, CNNs exhibited superior predictive accuracy. That may be due to the lack of apparent regularity in stock market price fluctuations and CNNs' stronger capabilities in extracting features of the current period. Based on the potential of CNNs, Hoseinzade et al. [63] introduced the CNNpred framework, designed to handle datasets from various markets. The CNNpred framework includes two variants: 2D-CNNpred and 3D-CNNpred. The 2D-CNNpred aims to build a universal model that maps historical data from any market to future price fluctuations, which can integrate market historical data and other relevant variables into a two-dimensional tensor for a custom CNN model to learn and predict from. The 3D-CNNpred, on the other hand, assumed that each market requires its specific model while allowing the model to utilize across markets data and merging multi-market data in a three-dimensional tensor form to train specific convolutional networks for each market. Experiments on multiple datasets, including S&P 500, NASDAQ, DJI, NYSE, and RUSSELL, have validated the effectiveness of the CNNpred framework. Dr. M. Durairaj and B. H. Krishna Mohan [64] approached financial time series' chaotic nature using chaos theory to model time series data, then input the modelling results into a CNN for initial predictions. To further enhance prediction accuracy, they employed polynomial regression to fit the prediction errors from the CNN, thus generating error prediction values. Combined with the initial forecast results, this error prediction forms a refined hybrid prediction model.

In the field of deep learning, as the network depth increases, traditional Convolutional Neural Networks (CNNs) often encounter the problem of gradient vanishing or exploding, which makes the networks difficult to train. To address this issue, the deep residual network (ResNet) emerged as a solution. ResNet introduces a unique residual learning framework that incorporates direct connections between layers, known as "skip connections", allowing signals to propagate directly from one layer to several others. This design not only prevents performance degradation in deep networks but also allows for a significant increase in network depth, achieving higher accuracy and better performance in various deep learning tasks. In ResNet, the input at layer l is denoted by x_l, which is processed by the residual function $F(x_l, W_l)$ involving weights W_l; this function can consist of multiple convolutional layers, resulting in the output x_{l+1}. The specific formula is as follows:

$$x_{l+1} = x_l + F(x_l, W_l) \tag{11}$$

Inspired by the WaveNet architecture, Lukas Börjesson and others in their research [65] integrated residual connections into their proposed neural network structure, ensuring that input information is preserved across multiple levels of the network and enhancing network sparsity. Experimental results show that, with the number of timesteps and filters held constant, the addition of residual connections is directly correlated with improved model performance. Another study by Zhao et al. [66] combined the LightGBM model with a wavelet denoised ResNet to predict foreign exchange rate changes over the following five intervals. As shown in Fig. 4, this model processes data every 30 timesteps through a sliding window mechanism, uses wavelet transforms to generate a 1 × 30 × 30 image matrix, and inputs it into a 50-layer ResNet with 16 residual blocks to extract high-level features. The results demonstrate that, compared to traditional CNNs, ResNet shows superior predictive performance, and the integration of wavelet denoising with LightGBM significantly enhances accuracy. Other studies, such as those by Liu [67] and Khodaee [68], have also incorporated residual structures in their network models, enhancing the transfer of data features and effectively mitigating performance degradation during network training.

Although vanilla CNNs can be used for time series prediction, they are limited by fixed convolutional kernel sizes and a limited receptive field, making it difficult to capture long-term dependencies in one-dimensional sequence data. To address this, Bai et al. [79] proposed the Temporal Convolutional Network (TCN) in 2018, which greatly expands the model's receptive field through extended convolutional kernels and dilated convolution techniques without the need for additional parameters. The dilated convolution formula is as follows:

$$F(s) = (\mathbf{x} *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{s-d\cdot i} \tag{12}$$

where $F(s)$ represents the output at time step s, x is the input signal, and f is the convolutional kernel. $*_d$ denotes dilated convolution operation, and k is the size of the kernel. $(\mathbf{x} *_d f)(s)$ is the weighted sum of the input x and the convolutional kernel f at time step s. Specifically, each element $f(i)$ of the kernel is multiplied by the corresponding element of the input sequence at the dilated position $\mathbf{x}_{s-d\cdot i}$, where the dilation factor d determines the spacing between these positions. This structural design of the Temporal Convolutional Network (TCN) enables it to capture long-range dependencies in sequence data efficiently, exhibiting superior capabilities compared to vanilla CNNs. Liu and colleagues proposed a Multi-channel Gated Temporal Convolutional Network (M-GTCN) [69], which utilizes multi-layer residual structures and TCN to construct channels, applies random Dropout to enhance generalization, and uses information fusion to enhance data representation, culminating in predictions through a fully connected layer. Experiments demonstrate that TCNs significantly outperform models like LSTM and GRU in univariate time series prediction tasks, and M-GTCN also achieves favourable results in multivariate time series prediction without additional time costs due to its multi-channel parallel structure. Deng and others [70] introduced the Knowledge-Driven Temporal Convolutional Network (KDTCN), a model that integrates structured events extracted from financial news and external knowledge from knowledge graphs with features extracted by TCN to predict and interpret stock trends. This model cleverly merges event embeddings with time series features, outperforming models that rely solely on numerical or textual data in experiments. Furthermore, Yao and colleagues [71] developed the MEMD-TCN model, which combines Multivariate Empirical Mode Decomposition (MEMD) with TCN. The MEMD algorithm decomposes multivariate time series into sub-sequences of different frequencies, which are then predicted by TCN. Experimental results show that MEMD-TCN excels in stock index prediction, surpassing other decomposition algorithms and AI models.

In addition, many scholars have attempted to combine Convolutional Neural Networks (CNN) with Recurrent Neural Networks (RNN), Long Short-Term Memory networks (LSTM), or Gated Recurrent Units (GRU) to develop hybrid models that leverage the advantages of two or even multiple types of networks. Kanwal et al. [72] proposed a hybrid predictive model that integrates Bi-directional CUDA Deep Neural Network LSTM (BiCuDNNLSTM) with 1D-CNN. The BiCuDNNLSTM utilizes its capability of bidirectional traversal of time series to extract more features from the same input dataset, thereby improving the accuracy of predictions. These features are then fed into a 1D-CNN for further mining and learning of rapid change features in stock data. Wu et al. [73] combined the advantages of CNN and LSTM to propose a Stock Sequential Array Convolution LSTM (SACLSTM) network. They transformed stock market parameters such as settlement prices, closing prices, and trading volumes, along with 30-time steps as coordinates, into a two-dimensional matrix suitable for 2D CNN processing that allows the CNN to learn spatial information in time series data like image analysis. Features extracted by the CNN layer are then fed into an LSTM layer to capture dynamic dependencies over time. Experimental results demonstrate the effectiveness of this method in stock market prediction tasks, showing superior performance compared to individual CNN or LSTM models, as well as traditional statistical methods. Similarly, Livieris et al. [74,75] have also applied the CNN-LSTM combination model to predict gold prices and cryptocurrencies.

Convolutional Neural Networks (CNNs) are good at capturing local data features, making them highly effective for tasks involving spatially localized information, such as image processing and some types of temporal data. However, due to their fixed local receptive fields and structural constraints, one-dimensional CNNs (1D-CNNs) have limitations in capturing global dependencies within time series data, which can affect their performance in complex time series prediction tasks. In tasks like stock market prediction, stock price data often exhibit high global correlation and dynamic changes, requiring models to understand and predict overall market trends and long-term dependencies. For applications in stock prediction, developing or improving 1D-CNNs to better capture and process these global characteristics becomes particularly important.

3.3. Stock forecasting models based on transformer

The attention mechanism [89] is designed to enhance the efficiency of neural networks in handling long-range sequence dependencies. This mechanism overcomes the constraint of RNNs, which must process data sequentially, and allows the model to compute dependencies between different positions in a sequence in parallel, thus effectively capturing long-term information. Self-attention architecture [90], an extension of the attention mechanism, models the relationships between different position feature vectors in an input sequence by computing their similarities, allowing for the capture of long-range dependencies. The mathematical computation of self-attention is generally represented as follows: given an input sequence $X = x_1, x_2, \dots, X_n$, where each x_i is a

Table 3
Stock forecasting models details based on transformer.

Author	Model	Method	Year	Research object	Evaluate metrics	Dataset
Yang [80]	HTML	Transformer, pre-trained WWM-BERT	2020	Price prediction	MSE	S&P500
Ramos [81]	Multi-Transformer	Transformer, Bagging	2021	Trend prediction	MAE, RMSE	S&P500
Wang [82]	Transformer	Transformer	2022	Price prediction	MAE, MSE, MAOE	CSI300, S&P500, HSI, NIKKEI225
Ma [83]	Stockformer	Transformer, GAT, CNN	2023	Trend prediction	MAE, RMSE, MAPE	S&P500
Mercier [84]	Hierarchical Refromer	Reformer, HiBERT	2022	Trend prediction	ROC-AUC, MCC, F1	TSX S&P60
Lu [85]	Informer	Informer	2023	Price prediction	MAE, RMSE, MAPE	HSI, NASDAQ, Tencent, AAPL
Ren [86]	Hybrid model	Informer, Encoder Forest	2023	Price prediction	MAE, MSE, RMSE, MAPE, MSLE, MAD, R2	China A-shares market
Liu [87]	PSO-Informer	Informer, PSO	2023	Price prediction	Accuracy, Precision, Recall, F1	SSE50, CSI300
Liu [88]	DMEformer	Informer, Reformer, Autoformer	2023	Price prediction	MAE, RMSE, MAPE, Accuracy	WTI crudeoil futures

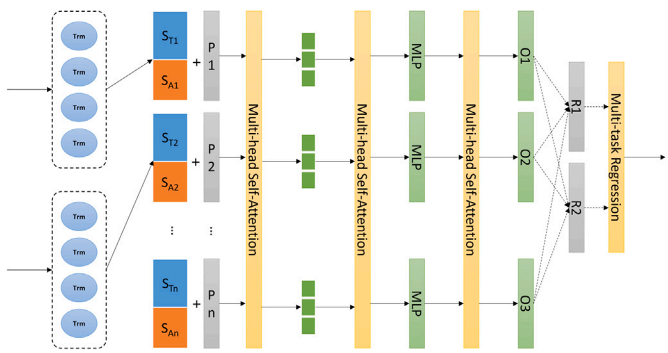

Fig. 5. HTML architecture.

feature vector at that position, the input is first transformed into queries (Q), keys (K), and values (V) using three sets of weight matrices. Then, by calculating the similarity between Q and all K (typically using dot product) and applying the softmax function, the weight distribution is obtained.

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V \tag{13}$$

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{14}$$

The matrices WQ, WK, and WV are learned parameters, and d_k is the dimensionality of the K vectors. The dot product is scaled by $\frac{1}{\sqrt{d_k}}$ to avoid an excessively large dot product that would cause the $softmax$ function to saturate in regions with smaller gradients. The $softmax$ function ensures that the weights for all positions sum to 1. Thus, the output for each position is the weighted sum of all values, where the weights reflect the similarity between the corresponding K and Q.

The Transformer model [91] introduces multi-head attention [92] based on self-attention. Its unique encoder–decoder architecture, with multi-head self-attention and feed-forward network components, enables the model to establish global dependency modelling without recursion while enhancing parallel processing capabilities and learning relationships between inputs across different representation subspaces. In time series forecasting, this capability allows the Transformer to capture various temporal patterns in parallel, from short-term fluctuations to long-term trends. In recent years, the Transformer and its variants have demonstrated superiority in analysing financial time series data, not only improving prediction accuracy but also significantly advancing processing speed and model interpretability. Table 3 provides the specific details of the reviewed Transformer methods.

Yang et al. [80] designed a Hierarchical Transformer-based Multi-task Learning method (HTML) to predict the S&P 500 stock market's future short- and long-term price fluctuations using textual and audio data from quarterly earnings calls. This method processes textual and audio data separately and converts them into corresponding token representations through Transformer encoding. Next, token-level feature fusion of these two modalities is conducted, followed by sentence-level Transformer encoding to capture cross-modal semantic and contextual information. The basic architecture of HTML is shown in Fig. 5. Ramos-Pérez et al. [81] introduced a Multi-Transformer network architecture that differs from traditional Transformer models by randomly selecting different subsets of training data and using bagging methods in conjunction with multiple multi-head attention mechanisms to generate the final output. This design improves the network's accuracy and effectively reduces the risk of overfitting. Unlike in natural language processing, Wang et al. [82] directly modelled daily closing price data using the Transformer rather than unstructured textual data, allowing the model to predict the exact stock index values rather than the market trend direction. Experiments in major global stock markets such as CSI 300, S&P 500, Hang Seng, and Nikkei 225 showed that the Transformer model significantly outperformed other traditional deep learning models and the "buy and hold" strategy in prediction accuracy and investment return analysis. Additionally, Ma et al. [83] introduced Stockformer, a cutting-edge deep learning framework optimized for swing trading, which enhances stock selection capability using the TopKDropout method while integrating STL decomposition and self-attention networks. Experimental results on the S&P 500 dataset revealed that Stockformer outperformed other industry models, exhibiting excellent performance in key prediction accuracy metrics (MAE, RMSE, MAPE) and achieving a 62.39% accuracy in market trend prediction.

To address the high memory and computational demands of Transformer models when processing large-scale datasets, researchers have developed variants like Reformer [93] and Informer [94]. Reformer reduces memory requirements significantly and optimizes long-sequence processing efficiency by incorporating reversible layers and locally sensitive hashing techniques. Informer, designed explicitly for time series forecasting, adopts the probability sparse transformation to reduce computational complexity, improving efficiency and performance in long-sequence data processing. These variants extend the applicability of Transformers and excel in large data environments. Inspired by HiBERT [95], Mercier et al. [84] developed a hierarchical Reformer model to analyse financial market documents or news to achieve high-quality long-sequence representations, successfully using them for predicting trading volume changes. Lu et al. [85] compared the performance of the Informer model with commonly used networks like LSTM, Transformer, and BERT in handling stock market data. They designed three comparative experiments across four market indices over 1-minute and 5-minute intervals. The results showed that Informer consistently achieved the best performance across all datasets. Transfer learning experiments also demonstrated that Informer has strong robustness and adaptability, effectively improving market forecasting performance. Ren et al. [86]

Table 4
Stock forecasting models details based on graph neural network.

Author	Model	Method	Year	Research object	Evaluate metrics	Dataset
Yang [97]	GGNN	GNN, GRU	2019	Event prediction	Accuracy, Precision, Recall, F1	Chinese listed companies News
Matsunaga [98]	GNN	GNN	2019	Trend prediction	RR, SR	NIKKEI225, Knowledge graph
Sawhney [99]	STHAN-SR	GNN, LSTM, Hypergraph	2021	Rank prediction	IRR, SR, NDCG	NYSE, NASDAQ, TSE
Cheng [100]	MAGNN	GNN, Transformer, pre-trained BERT	2022	Price prediction	Micro-F1, Macro-F1 Weighted-F1	China A-shares market, Knowledge graph
Zhao [101]	DANSMP	GNN, NTN, GRU	2022	Trend prediction	Accuracy, Precision, Recall, F1	CSI100, CSI300, Knowledge graph
Chen [102]	GCN	GCN, LSTM	2018	Trend prediction	Accuracy	CSI300
Feng [103]	GCN	GCN, LSTM, RSR	2019	Rank prediction	MSE, MRR, IRR, RR	NASDAQ, NYSE
Gao [104]	TRAN	GCN, LSTM	2021	Rank prediction	MSE, MRR, IRR	NASDAQ, NYSE
Chen [105]	GC-CNN	GCN, CNN	2021	Trend prediction	Accuracy, Precision, Recall, F1, TMoney, AAR, SR	China A-shares market
Li [106]	Chart GCN	GCN, CNN	2022	Trend prediction	Accuracy, Precision, F1	SZ50, CSI300
Ma [107]	AFHGN	GCN, GRU, Fuzzy clustering, Hypergraph	2022	Rank prediction	NDCG, SR, IRR	China A-shares market
Song [108]	MGAR	GCN, LSTM	2023	Rank prediction	MSE, MRR, IRR, SR, MDD	NASDAQ, NYSE
Peng [109]	RTGCN	GCN, External Attention	2023	Trend prediction	Accuracy, Precision, F1, MCC	CSI100, CSI300, Knowledge graph
Sawhney [110]	MAN-SF	GAT	2020	Trend prediction	Accuracy, F1, MCC	–
Hsu [111]	FinGAT	GAT, GRU	2021	Trend prediction	MRR, Precision, Accuracy	Taiwan Stock, S&P500, NASDAQ
Feng [112]	RA-AGAT	GAT, Attn-LSTM	2022	Trend prediction	Accuracy, MSE, MRR	CSI300
Ma [113]	GAT	GAT, LSTM	2023	Rank prediction	ACC, MCC, AR, SR	China A-shares market
Lei [114]	DR-GAT	GAT, GRU	2024	Rank prediction	MSE, MRR, IRR	NASDAQ, NYSE

In this table, **Event prediction** assesses particular occurrences that could influence stock prices and forecasts their potential market impact. **Rank prediction** estimates and orders stocks based on their expected future performance, typically aiming to identify potential high-performing assets for investment purposes.

proposed a model integrating Encoder Forest (EF) with Informer. To mitigate the impact of long-sequence noise on stock predictions, they decomposed the original data into high-frequency signal components (CD) and low-frequency signal components (CA). Liu et al. [87] optimized the parameters of the Informer network using a particle swarm optimization algorithm and developed a long-term stock price sequence forecasting method called PSO-Informer. Liu et al. [88] introduced the Dynamic Model Ensemble Transformer (DMEformer), which integrates three different Transformer variants — Autoformer, Reformer, and Informer — using a dynamic ensemble strategy to predict US oil futures data. Investigating the effectiveness of Transformers for time series forecasting, Zeng et al. [96] raised a crucial question: "Are transformers effective for time series forecasting?" They pointed out that most Transformer models fail to capture temporal relationships in long sequences effectively, and when the look-back window is expanded, prediction errors do not decrease but rather increase. To verify this viewpoint, Zeng et al. designed a simple linear model, LTSF-Linear, and compared it with several complex Transformer-based models. The results indicated that LTSF-Linear outperformed existing complex Transformer models across nine widely used benchmark datasets in all test scenarios. Therefore, whether and how to use Transformers more effectively remains a significant proposition for future researchers to explore.

Transformers utilize self-attention mechanisms to solve long-term dependency issues, thereby increasing efficiency and accuracy in processing sequential data, particularly achieving significant breakthroughs in natural language processing and related fields. However, for highly long sequences, Transformers may encounter challenges with high computational and memory demands, limiting their practicality in large-scale applications. In stock market prediction, the length and complexity of time series data that need to be processed often exceed conventional ranges, which may put significant pressure on the resources and processing speed of Transformer models. Therefore, developing or optimizing Transformer architectures to efficiently handle long sequences and reduce resource consumption is critical to adapting to the needs of stock prediction.

3.4. Stock forecasting models based on Graph Neural Network

Graph Neural Networks (GNNs) [115] are designed explicitly for processing graph-structured data, allowing for aggregating and updating information across nodes to capture complex relationships and

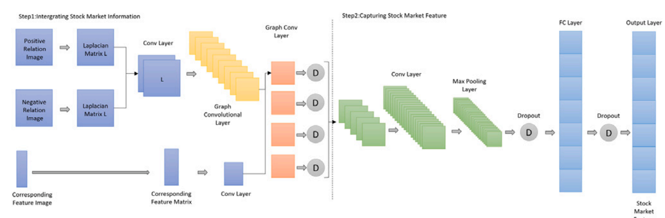

Fig. 6. IGCN architecture.

dependencies. GNNs update node states by integrating their features with those of their neighbours, effectively learning representations within the graph. In stock forecasting, GNNs demonstrate unique advantages. The stock market can be viewed as a complex graph where nodes represent different stocks and edges signify their relationships. By learning the features of these nodes and edges, GNNs can effectively capture the dynamic associations and influences between stocks, thereby forecasting future stock price movements. This approach is particularly suitable for multi-asset prediction tasks, offering deeper insights into market dynamics and higher predictive accuracy than traditional time series-based models, providing a more comprehensive understanding of the market. Table 4 provides the specific details of the reviewed GNN methods.

Yang et al. [97] proposed a method based on Gated Graph Neural Networks (GGNN) to capture complex relationships between event graphs. The model uses historical event chains as input and is trained through a cloze task to predict upcoming events. Additionally, they incorporated financial news as a supplementary information source to address multiple interpretations of the same financial event. Matsunaga et al. [98] explored how to integrate knowledge graphs with graph neural networks to enhance the accuracy and generalization of stock market predictions. They used graph neural networks to simulate the way investors analyse market connections and directly incorporated enterprise-related knowledge graphs into the stock market prediction model. A long-term rolling window analysis of about 20 years in the Nikkei 225 market showed that their model's return and Sharpe ratio were 29.5% and 2.2 times higher than the market benchmark, respectively, an increase of 6.32% and 1.3 times compared to traditional

LSTM models. Sawhney et al. [99] redefined the stock prediction problem as a ranking learning problem and proposed STHAN-SR. This model utilizes a hypergraph and a temporal Hawkes attention mechanism to model stock dependencies and prices over time, achieving good performance across multiple datasets. Given that stock price trends reflect complex market conditions at different diffusion speeds, including historical price sequences, media news, and related events, Cheng et al. [100] proposed a multimodal graph neural network (MAGNN) to learn from these multimodal inputs. This network is built on a financial knowledge graph, with nodes composed of source data and edges representing relationships. To enhance the model's interpretability, the researchers used a two-stage attention mechanism for joint optimization, allowing end-users to learn inter-modal and intra-modal information. Specifically, the authors first extracted relationships of linked entities from original news and then stored this information in a financial knowledge graph (FinKG) containing 5.26 million entities and 6.93 million relationships. Zhao et al. [101] first attempted to use a heterogeneous graph neural network (GNN) to explore the spillover effects of a dual-type hybrid relationship knowledge graph in the stock market. They proposed an innovative dual-attention network, DANSMP, to learn the features of stock momentum spillover in the dual-type hybrid relationship knowledge graph (MKG) for stock prediction.

Graph Convolutional Networks (GCN) [116] is a variant of Graph Neural Networks (GNN) that extend the concept of Convolutional Neural Networks (CNN) to graph-structured data, effectively achieving feature learning on graph nodes. GCN operates by applying convolution operations to each node in the graph, using the features of adjacent nodes to update the current node's feature representation. This method allows each node to capture information about its neighbourhood structure, thus creating a global graph representation. Chen et al. [102] developed a joint prediction model based on GCN, combining information about target companies and their associated companies to predict stock prices. By constructing a graph neural network that includes all relevant companies and applying node embedding techniques, the model learns the distributed representation of each company, thereby more comprehensively capturing market dynamics. Feng et al. [103] proposed a new deep learning solution named Relational Stock Ranking (RSR), which includes a network structure called temporal graph convolution to address stock ranking issues in a time-sensitive manner. Testing on NYSE and NASDAQ data showed a 115% improvement in return rates compared to baselines. Gao et al. [104] introduced a graph-based stock recommendation method called Time-aware Relational Attention Network (TRAN), which recommends stocks based on return ratio ranking. The time-aware relational attention mechanism aims to capture the evolving correlation strengths between stocks over time through interactions between historical sequences and stock description documents. Chen Wei et al. [105] proposed using an improved Graph Convolutional Network (IGCN) and a Dual Convolutional Neural Network (Dual-CNN) to create a Graph Convolutional Feature Convolutional Neural Network (GC-CNN) that integrates stock market information and individual stock features. By selecting a few stocks that represent the entire market, they built a stock network to capture the market's topological structure and designed feature matrices to represent the characteristics of each stock in the market. The architecture of IGCN is shown in Fig. 6. The IGCN framework is divided into two main components: integrating stock market information and capturing stock market features. Its primary objective is to extract stock market features by leveraging positive and negative relational images and their corresponding feature images. Stock market and individual stock information are processed through IGCN and Dual-CNN, respectively. Experimental results show that this GC-CNN-based method can achieve higher profits than other methods. Li et al. [106] proposed a strategy to transform technical charts into graphs to address the limitations of traditional technical analysis, which only considers specific chart similarities, and used a Graph Convolutional Network (GCN) to compare the similarity of these graphs. Ma et al. [107] proposed the Attribute-Driven Fuzzy Hypergraph Network (AFHGN) to address issues of improper aggregation operations and unreasonable hyperedge constructions in hypergraph stock analysis. The network utilizes fuzzy clustering to construct node community matrices, introduces attribute-driven gate units, and establishes trend weights to simulate the influence of stocks in real markets. Extensive experiments have shown that this method outperforms the state-of-the-art (SOTA) methods at that time. Song et al. [108] proposed the Multi-Relational Graph Attention Ranking (MGAR) network, which employs adaptive learning mechanisms to aggregate multiple graphs, forming effective relation embeddings. Based on captured price trend embeddings, the MGAR model provides a ranked list of future returns and selects the top-performing K stocks for trading to maximize investment returns. Considering that GCN-based stock prediction methods typically only extract signal representations of individual stocks at the feature extraction stage and ignore real-time interactions between companies, Peng et al. [109] proposed a Relationship Type-Guided Graph Convolutional Network (RTGCN) with an External Attention (EA) module for stock price movement prediction (SPMP). This external attention mechanism shares the memory of price and sentiment signals among companies, thereby enhancing the model's predictive power.

In addition to GCNs, Graph Attention Networks (GATs) [117] represent another significant variant of GNNs. GATs incorporate attention mechanisms to dynamically determine the importance of neighbouring nodes during the node updating process, which allows GATs to finely consider the contributions of surrounding nodes when constructing new feature representations for nodes, thereby capturing the structural information of graphs more effectively. In stock forecasting, GATs can analyse complex relationships between various economic entities, such as competition and cooperation between companies, supply chain connections, or the impact of macroeconomic factors. Sawhney et al. [110] developed a multipronged attention network called MAN-SF, which uses a layered temporal model to effectively integrate complex temporal signals from financial data, social media, and inter-stock relationships. The FinGAT model developed by Hsu et al. [111] includes three main components: feature learning at the stock level, industry level, and multitask learning. A distinctive feature of FinGAT is that it does not rely on predefined stock relationships but uses Graph Attention Networks to explore potential interactions between stocks and industries automatically. Feng et al. [112] proposed a relation-aware dynamic attribute graph attention network (RA-AGAT), which combines temporal features encoded by a time series module and global information provided by a stacked graph attention network (GAT). Extending the attention mechanism from node features to topological information, this model incorporates the stock correlations into the message-passing process. To address the lag and overfitting issues of static stock graphs, Ma et al. [113] introduced a dynamic graph construction module on top of GNN, utilizing MoDis and DGLSTM to build dynamic graphs. Additionally, they proposed a novel stock distance algorithm based on topic detection. Experiments on data from 4503 stocks in the Chinese A-share market demonstrated that their method of constructing dynamic graphs can promptly respond to changes in stock relationships, significantly enhancing stock trend prediction performance. In order to learn latent relationships between stocks, Lei et al. [114] proposed a Dynamic Routing Graph Attention Network (DR-GAT) for stock recommendation. Their proposed Relation Graph Router (RGR) routes each stock to an optimal relation graph based on its volatility characteristics.

Overall, Graph Neural Networks (GNNs) perform excellently in handling and analysing complex node relationships in the stock market, effectively capturing information flow and interactions within stock market networks. However, GNNs are highly sensitive to the topology of graphs, necessitating high-quality and complete graph data to ensure prediction accuracy and stability. In stock market prediction, the quality of the graph directly impacts model performance, as the dynamic changes and complexity of the stock market network require GNNs to adapt to these changes and accurately reflect the actual relationships between nodes. Therefore, for GNN stock prediction, ensuring high-quality graph data input is crucial.

Table 5
Stock forecasting models details based on generative adversarial network.

Author	Model	Method	Year	Research Object	Evaluate Metrics	Dataset
Zhou [118]	GAN-FD	GAN, CNN, LSTM	2018	Price prediction	DPA, RMSRE	CSI300
Mariani [119]	PAGAN	GAN, CNN	2019	Trend prediction	RR, SR	Yahoo Finance
Zhang [120]	GAN	GAN, LSTM, MLP	2019	Price prediction	MAE, RMSE, MAPE, AR	S&P500, CSI, NASDAQ
Wu [121]	GAN	GAN, LSTM, PLR	2023	Trend prediction	Investment (CR), SR, and winning percentage (WPTC)	TAIEX
Jadhav [122]	GAN	GAN, LSTM, MLP, (Naive Bayesian Model) NBM	2021	Price prediction	MAE, RMSE, MAPE, Accuracy	S&P500, CSI300, DJI, BSE, NASDAQ, Twitter Data
Muthukumar [123]	ST-GAN	GAN, NBM	2021	Price prediction	RMSE, NRMSE	Yahoo Fiance
Koshiyama [124]	cGAN	cGAN	2021	Trend prediction	RMSE, SR, CR	S&P500, FTSE100, DIJA

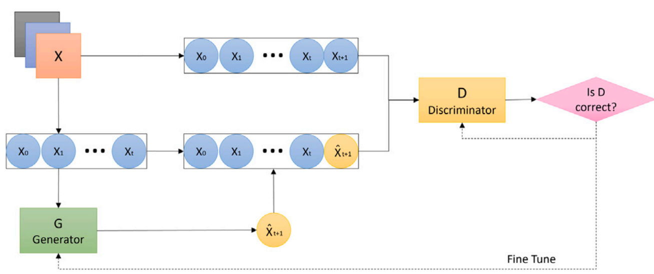

Fig. 7. GAN architecture.

3.5. Stock forecasting models based on Generative Adversarial Network

Generative Adversarial Networks (GANs) [125] were proposed by Ian J. Goodfellow in 2014 as an innovative deep learning framework composed of a generator and a discriminator. Through an adversarial training mechanism, the generator produces realistic data samples, while the discriminator strives to distinguish between generated samples and real samples [126]. In the field of stock forecasting, GANs are used to augment datasets, allowing predictive models to more comprehensively learn and simulate market dynamics, particularly excelling in simulating rare or extreme market events. Table 5 provides the specific details of the reviewed GAN methods.

Zhou et al. [118] was the first to apply Generative Adversarial Networks (GANs) to stock market forecasting, developing a universal adversarial training framework based on LSTM and CNN, named GAN-FD. They used public indices provided by trading software as input and employed a rolling partition method to analyse the impact of model periodic updates on forecasting performance. Mariani et al. [119] introduced PAGAN, a GAN-based portfolio analysis method that models market uncertainties driving future trends directly, embedding nonlinear interactions between different assets to provide new insights for investment management. As shown in Fig. 7, Zhang et al. [120] used LSTM as the generator and MLP as the discriminator to predict daily stock closing prices. Wu et al. [121] proposed a GAN method that integrates Piecewise Linear Representation (PLR) to learn three types of trading behaviours in the stock market: buying, selling, and holding. The PLR technique constructs sequences that include specific trading actions, thus guiding the discriminator with actual trading strategies. Experimental results showed that this framework outperformed LSTM, effectively improving the accuracy of trading predictions. Jadhav et al. [122] analysed various financial forecasting algorithms and observed their performance on their respective datasets based on various evaluation metrics. Inspired by [120], they proposed an innovative financial analysis system that combines a Naive Bayes classifier for sentiment analysis of financial news with a Long Short-Term Memory network (LSTM) as the generator and a Multi-Layer Perceptron (MLP) as the discriminator to predict stock prices. Muthukumar et al. [123] introduced ST-GAN, using GAN to adversarially learn the temporal correlation between financial news and financial data, also employing Naive Bayes for sentiment analysis of financial texts. Experiments on Yahoo Finance data showed that the proposed method achieved significant improvements over the baseline.

Conditional Generative Adversarial Networks (cGANs) [127] are an extension of GANs that incorporate additional conditions or labels when generating data, enhancing the precision and relevance of the outputs. In cGANs, both the generator and discriminator receive conditional information, with the generator aiming to produce data that meets specific conditions while the discriminator evaluates the authenticity and accuracy of the conditioned data. This makes cGANs particularly suitable for applications requiring fine control over generated content, significantly enhancing the utility of GANs. Koshiyama et al. [124] utilized cGAN to calibrate and aggregate trading strategies, testing on 579 datasets comprising stocks, futures, and currency data. The study found that when traditional techniques like Bootstrap and Bagging performed moderately, cGAN could provide excellent performance, serving as an effective alternative.

Generative Adversarial Networks (GANs) generate high-quality and diverse data samples through adversarial training, holding the potential for simulating market data such as stock price fluctuations. However, the training process of GANs can be unstable, particularly in complex stock prediction tasks, requiring meticulous tuning and handling to achieve reliable results. In stock market prediction, this instability poses a challenge, as the uncertainty of stock price data and rapid changes in market conditions can make model training more difficult. Therefore, the key to improving the effectiveness and stability of GANs in stock prediction lies in optimizing training strategies and parameter settings to ensure that the model can effectively learn and simulate complex market dynamics.

3.6. Stock forecasting models based on Large Language Model

In recent years, Large Language Models (LLMs) such as BERT and GPT have demonstrated remarkable performance in natural language processing tasks, especially in text generation, semantic understanding, and sentiment analysis [136,137]. These models' capabilities have made them particularly valuable in the financial sector, notably in stock forecasting. By analysing text data from news, financial reports, and social media, these models can capture subtle changes in market sentiment and public perception that influence stock market fluctuations. Combined with historical price data, these models can perform complex time series analysis and accurate stock price predictions, revealing trends in market changes. Among the most renowned LLMs are BERT and GPT; their basic architectures are shown in Fig. 8. Table 6 provides the specific details of the reviewed LLM methods.

BERT (Bidirectional Encoder Representations from Transformers) [138] is an advanced variant based on the Transformer architecture, designed to capture bidirectional contextual information in language. As a representative of current LLMs, BERT employs the encoder part of the Transformer along with innovative pre-training techniques such as the Masked Language Model (MLM) and Next Sentence Prediction (NSP), which effectively train the model to understand and predict complex relationships in language. Yang et al. [80] utilized a pre-trained WWM-BERT to process text information and generate token embeddings. To

Table 6
Stock forecasting models details based on large language model.

Author	Model	Method	Year	Research object	Evaluate metrics	Dataset
Zhao [128]	Multi-layer BERT	BERT	2022	Trend prediction	Accuracy, F1	–
Sousa [129]	BERT	BERT	2019	Trend prediction	Accuracy, Precision, Recall, F1	DJI
Sidogi [130]	FinBERT	BERT, LSTM	2021	Price prediction	MAE, RMSE	NASDAQ Index, News and Text
Hiew [131]	BERT	BERT, LSTM	2022	Trend prediction	Precision, Recall, F1	Tencent CCB, Ping An
Huang [132]	FinBERT	BERT	2023	–	Accuracy, Precision, Recall, F1	NASDAQ
Garza [133]	TimeGPT	TimeGPT	2023	Price prediction	rMAE, rRMSE	–
Jin [134]	Time-LLM	LLaMa	2023	Price prediction	MAE, MSE, SMAPE, MAPE, MASE, OWA	–
Yu [135]	Fine-tuned LLaMa	LLaMa	2023	Trend prediction	MSE	NASDAQ

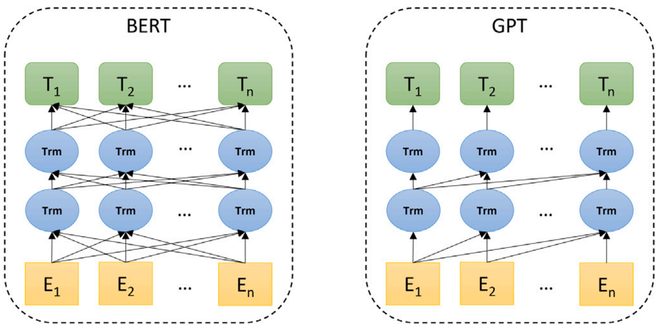

Fig. 8. Based architectures of BERT and GPT.

address the limitation of BERT in processing long texts and feature extraction, Zhao et al. [128] proposed an improved BERT model that uses a sliding window technique to segment long texts into multiple paragraphs to increase the sample size, then aggregates the central themes of these short text paragraphs. Additionally, this model extracts features from each layer of BERT and employs a multi-layer feature ablation strategy to select the most effective information. Experimental results have shown that this method significantly improves the accuracy of identifying themes in stock commentary.

The sentiment in the financial markets also reflects some of the reasons for market changes. Sousa et al. [129] used a BERT model to perform sentiment analysis on stock market news articles to predict future trends in the Dow Jones Industrial Average (DJI). The research team manually categorized these articles as positive, neutral, or negative, then fed the labelled data into a self-supervised pre-trained BERT model optimized for a wide range of general domain documents. This approach improved the accuracy of stock index trend predictions through sentiment analysis. Sidogi et al. [130] pointed out that more than relying on historical price data is needed for future stock price predictions. Therefore, they employed a fine-tuned FinBERT model [139] to encode news information and classify emotions in the news, then input these emotional data along with historical price data into the prediction model. The experimental results showed that this approach, which combines news sentiment analysis, significantly outperforms traditional baseline models. Hiew et al. [131] also used a BERT model to analyse emotions in financial news and developed a text-based financial sentiment index for three popular stocks on the Hong Kong Stock Exchange. By integrating information from various channels, including text data, options, and market data, they designed a comprehensive financial sentiment analysis framework that effectively distinguishes between the emotions of individual investors and institutional investors. Huang et al. [132] developed a model specifically for textual analysis in the financial domain named FinBERT. Although it shares the name, the structure of this model differs in that it is based on the original BERT model but includes an ESG issue identification module and has been specially optimized for small sample data to enhance performance in specific application scenarios. FinBERT, through domain-specific pre-training with extensive financial sector texts, gains a deep understanding of financial terminology and context. Moreover, by weighting financial vocabulary and optimizing the sentiment classification process, FinBERT significantly improves the accuracy of sentiment analysis and professional terminology handling.

Another noteworthy large language model is GPT (Generative Pre-trained Transformer) [140], which unlike BERT's bidirectional encoder structure, is based entirely on a decoder architecture, focusing on generating coherent text. It primarily learns language patterns through unsupervised pre-training from a vast corpus of text, then adapts to various downstream tasks through a supervised fine-tuning process. This combination of pre-training and fine-tuning strategies significantly enhances GPT's flexibility and applicability, making it excel in text generation tasks and particularly suitable for financial forecasting. Garza et al. [133] introduced TimeGPT-1, the first pre-trained foundational large model explicitly designed for time series data, capable of generating accurate predictions on a variety of unseen datasets. TimeGPT, based on the Transformer design, captures the temporal dependencies of data through self-attention mechanisms and uses historical data windows and local position encodings to predict future values. Each layer of the encoder and decoder includes residual connections and layer normalization, enhancing training stability and prediction accuracy. Trained on over 100 billion data points, TimeGPT covers multiple domains, including finance, economics, demographics, healthcare, weather, IoT sensors, energy, network traffic, sales, transportation, and banking. When compared against various statistical models and advanced deep learning methods, TimeGPT consistently ranks among the top performers, whether the predictions are monthly, weekly, daily, or hourly.

In addition, several emerging large models with different architectures have also started to gain prominence in stock prediction. Jin et al. [134] proposed a time-series reprogramming large model framework named Time-LLM. This framework retains the backbone language model while reprogramming the input time-series data using text prototypes before feeding the reprogrammed features into the frozen large model to align the two modalities. Additionally, they introduced Prompt-as-Perfix (PaP) to enrich contextual information. This strategy of reprogramming time-series data makes the input features more compatible with the large language model's input format. Experimental results indicate that Time-LLM outperforms specialized prediction models and excels in few-shot and zero-shot learning tasks. Yu et al. [135] fine-tuned the open-source large model LLaMa [141] using instructions and conducted stock price time-series prediction on the NASDAQ-100 index. The experimental results show that, in most cases, the chain-of-thoughts technique helps improve prediction performance, and the fine-tuned model surpasses the classic ARMA-GARCH and gradient-boosting tree models.

Large Language Models (LLMs) like BERT and GPT understand and generate natural language, offering powerful tools for analysing and predicting financial news and reports based on text. In addition to text data, these models can also process different types of input data, such as time series data, images, and tables, enabling them to provide a more comprehensive market analysis. However, these models may overly rely on training data distribution and struggle with novel or anomalous financial events. Additionally, they typically require extensive training resources, and their black-box nature limits transparency in the decision-making process, posing challenges for investment decisions.

4. Datasets and evaluation metrics

In the field of stock forecasting, the richness of datasets and the accuracy of evaluation metrics form the foundation for research and are key to assessing model efficacy. A deep understanding of the sources, structures, and unique characteristics of different datasets is crucial for building effective prediction models. Choosing appropriate evaluation metrics reflects model performance and guides optimization directions. This section will explore the commonly used datasets and evaluation metrics in stock forecasting research, aiming to provide researchers with a comprehensive reference framework that deepens theoretical understanding and offers practical guidance for decision support and model adjustment in practical operations.

4.1. Datasets

The data types involved in the field of stock forecasting are complex and varied, encompassing multi-dimensional information from time series data to text data, social media information, audio materials, and even knowledge graphs [142]. Each data type provides unique insights for prediction models, and the characteristics and purposes of these data types are detailed below:

- **Time Series Data:** This type of data typically includes stock historical prices, trading volumes, and opening and closing prices and is the most basic and crucial data source for stock forecasting.
- **Text Data:** Text content from news reports, financial statements, company announcements, and research reports can be analysed using natural language processing techniques, providing valuable qualitative insights into market sentiment and corporate performance.
- **Social Media Data:** The plethora of information on social media platforms like Weibo, Twitter, and stock forums reflects public sentiment and market reactions to specific events, making it an indispensable data source for modern financial predictions.
- **Audio Data:** Audio materials such as earnings call conferences and market analyst speeches can be mined for deeper insights affecting market trends through voice recognition and sentiment analysis technologies.
- **Knowledge Graphs:** By integrating various entities (such as companies, industries, and products) and their interrelationships, knowledge graphs provide structured depth information on competition and collaboration between companies and industry structures.

Table 7 organizes common data sources for stock forecasting, including abbreviations of different stock markets and exchanges, corresponding regions, and detailed explanations, providing readers with a clear index of data sources.

4.2. Evaluation metrics

In the field of stock forecasting, the choice of evaluation metrics closely depends on the specific goals of the research. For methods aimed at precisely estimating market prices that fluctuate over time, regression evaluation metrics are crucial in measuring model performance. These include Mean Squared Error (MSE), Root Mean Squared Error ($RMSE$), Mean Absolute Error (MAE), Mean Absolute Percentage Error ($MAPE$), and the Coefficient of Determination (R^2). These metrics quantify the degree of deviation between the model's predictions and the actual market prices. Another category of methods focuses more on capturing directional trends in market changes, such as stock price movements, rather than precise values. For such methods, classification evaluation metrics such as $Accuracy$, $Precision$, $Recall$, and the $F1Score$ provide a robust measure of model performance, especially in identifying market trends. Evaluation metrics for ranking tasks are standard in recommendation systems and information retrieval and apply to stock ranking predictions. These include Mean Reciprocal Rank (MRR), Discounted Cumulative Gain (DCG) and Normalized Discounted Cumulative Gain ($NDCG$). Lastly, the core goal of stock forecasting is to guide investors in identifying profitable patterns in a volatile market. Therefore, performance metrics oriented towards returns, such as the Sharpe Ratio (SR) and Return Ratio (RR), are crucial because they directly reflect the model's profitability in real market scenarios. The following section will detail the above evaluation metrics:

4.2.1. Mean squared error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{15}$$

MSE calculates the average of the squares of the differences between the actual values y_i and the predicted values $\hat{y}_i$. It measures the average size of the prediction errors, with a smaller MSE indicating higher model accuracy.

4.2.2. Root mean squared error

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{16}$$

$RMSE$ is the square root of MSE, representing the error in the same units as the actual observations, making it easier to interpret.

4.2.3. Mean absolute error

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{17}$$

MAE measures the average of the absolute differences between actual and predicted values. Compared to MSE, MAE is less sensitive to outliers.

4.2.4. Mean absolute percentage error

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{18}$$

$MAPE$ expresses the prediction error as a percentage of the actual values, commonly used to measure the relative error of predictions.

4.2.5. Coefficient of determination

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{19}$$

R^2 indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. It is often used to gauge the goodness of fit of a model. The closer the R^2 value is to 1, the more effectively the model can predict the data.

4.2.6. Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

$Accuracy$ is the ratio of correctly predicted observations to the total number of observations, where TP (True Positives), TN (True Negatives), FP (False Positives), and FN (False Negatives) are the counts of true positive, true negative, false positive, and false negative predictions, respectively.

Table 7
Stock market and exchange information.

Abbreviation	Country	Type	Detail
CSI 100	China	Index	China Securities 100 Index
CSI 100	China	Index	China Securities 100 Index
SSE 50	China	Index	Shanghai Stock Exchange 50 Index
HSI	China	Index	Hang Seng Index
STI	Singapore	Index	Straits Times Index
NIKKEI 225	Japan	Index	Nikkei 225 Index
NIFTY 50	India	Index	National Stock Exchange of India 50 Index
KOSPI 200	Korea	Index	Korea Composite Stock Price 200 Index
STOXX50	Europe	Index	Eurozone countries STOXX 50 Index
RTSI	Russia	Index	Russian Trading System Index
FCHI	France	Index	France CAC 40 Index
DAX	Germany	Index	Deutscher Aktienindex (German Stock Index)
FTSE 100	UK	Index	Financial Times Stock Exchange 100 Index
DJI/DJIA	USA	Index	Dow Jones Industrial Average Index
NASDAQ	USA	Index	NASDAQ Composite Index
RUSSELL	USA	Index	Russell Indexes
S&P 500	USA	Index	Standard & Poor's 500 Index
SSE	China	Exchange	Shanghai Stock Exchange
SZSE	China	Exchange	Shenzhen Stock Exchange
HKEX	China	Exchange	Hong Kong Stock Exchange
NSE	India	Exchange	National Stock Exchange of India
LON	UK	Exchange	London Stock Exchange
NYSE	USA	Exchange	New York Stock Exchange

4.2.7. Precision

$$Precision = \frac{TP}{TP + FP} \tag{21}$$

$Precision$ is the ratio of correctly predicted positive observations to the total predicted positives. It reflects how many of the identified positive instances are actually positive.

4.2.8. Recall

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

$Recall$, or sensitivity, is the ratio of correctly predicted positive observations to all actual positives. It measures how well the model identifies all relevant instances.

4.2.9. F1 score

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{23}$$

The $F1$ Score is the harmonic mean of Precision and Recall. It is a balance between the precision and recall of the model, providing a measure of accuracy and comprehensiveness.

4.2.10. Mean reciprocal rank

$$MRR = \frac{1}{Q} \sum_{q=1}^{Q} \frac{1}{rank_q} \tag{24}$$

The MRR measures the effectiveness of a ranking task for individual queries, where Q donates the total number of queries, $rank_q$ is the real rank of the predicted top-1 stock in the ground-truth on the qth testing day.

4.2.11. Discounted cumulative gain

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i + 1)} \tag{25}$$

DCG quantifies the cumulative relevance of results in a ranked list. Here, k is the length of the truncated list of ranks, and rel_i is the relevance score of the result at position i.

4.2.12. Normalized discounted cumulative gain

$$nDCG_k = \frac{DCG_k}{IDCG_k} \tag{26}$$

NDCG normalizes DCG to facilitate comparisons across different queries and datasets.

4.2.13. Sharpe ratio

$$SR = \frac{R_p - R_f}{\sigma_p} \tag{27}$$

The SR measures the adjusted return for risk, where R represents the expected return of the portfolio, R_f is the risk-free return ratio, and σ_p is the standard deviation of the portfolio. This ratio helps understand how much excess return is received for the extra volatility of holding a riskier asset compared to a risk-free asset.

4.2.14. Return ratio

$$RR = \frac{FinalValue - InitialValue}{InitialValue} \tag{28}$$

The RR calculates the growth ratio between the initial and final values of an investment. It measures the total return of an investment relative to its initial cost, providing a straightforward metric of investment performance over a period.

5. Open problems and promising directions

In reviewing stock forecasting models, we have thoroughly analysed the design, characteristics, and performance of various models. Although current stock forecasting technologies have achieved significant accomplishments, existing research still faces numerous challenges, and the prospects in this field remain broad. This chapter will highlight these unresolved issues and explore promising directions, aiming to provide researchers with a clear problem framework and stimulate innovative thinking to promote continuous progress and innovation in the field of stock forecasting.

5.1. Insufficient data representation capacity

One-dimensional time series data refers to sequences arranged in chronological order, where each data point represents an observation at a specific time [143]. This form of data is common in many practical applications, such as meteorological records, stock market prices, and electrocardiograms. In these applications, time series data helps us understand how events change over time and predict future trends.

In the stock market, one-dimensional time series data primarily includes indicators such as stock prices and trading volumes. These data are continuously recorded but are often noisy and influenced by external events, making them easy to anomalies and errors. Additionally, stock forecasting models often perform poorly when faced with unseen data due to their lack of generalizability. Recent studies, such as [144–146], have utilized self-supervised learning as a pre-training task for stock market prediction, compensating for the lack of labelled data and allowing models to learn from a vast amount of unlabelled financial data. This approach enhances the adaptability and robustness of the model and effectively explores hidden patterns and structures in the data. Zhang et al. [147] believe that self-supervised learning has tremendous potential in data augmentation, robustness analysis, and benchmark assessment. Therefore, time series analysis based on self-supervised learning shows broad research prospects, whether in mining existing strategies or exploring new methods.

The features of one-dimensional time series data are relatively simple, primarily focusing on price fluctuations, leading to insufficient overall feature representation capability. Inspired by the field of computer vision, Wang et al. [148] first proposed using the GAF transformation to encode time series into images, allowing machines to "visually" recognize, classify, and learn structures and patterns. This method of transforming time series into images not only retains the temporal and spatial information of the data but also significantly enhances the data's representational power. Specifically, the GAF transformation converts one-dimensional data into a two-dimensional image form by calculating the inner product relationships between points in the time series, effectively using the time and spatial dependencies of known data points to predict missing values. Subsequent studies, such as [149–151], have adopted similar strategies to enhance data representation.

Multimodal learning [152] integrates information from different data sources, such as text, images, audio, and traditional numerical time series data, providing a more comprehensive market perspective. This learning approach is particularly suitable for stock prediction, as various factors, including economic indicators, corporate news, and social media sentiment, influence market fluctuations. By merging and learning from multiple different data modalities, not only can the model's understanding of market dynamics be enhanced, but also its robustness when faced with incomplete or low-quality data. Multimodal learning technologies have been widely applied in fields such as healthcare, autonomous driving, and robotics and are gradually showing potential in the stock prediction domain [100,153].

In summary, the representational capability of raw one-dimensional time series data is limited, making it difficult to capture complex market dynamics fully. To enhance data representation, we can start from the following aspects: (1) Self-supervised learning: deeply explore the inherent patterns and structures of data through self-supervised learning; (2) Time series to image encoding: convert one-dimensional time series data into two-dimensional image form to preserve temporal and spatial dependencies; (3) Multimodal learning: integrate various data sources, such as text and images, to provide a more comprehensive market perspective for the model. Through these methods, we can effectively expand the representational ability of the original data, laying a more solid foundation for subsequent modelling and prediction work.

5.2. Challenges in long-term dependencies modelling

Long-term dependencies modelling has always been a core issue in time series analysis, particularly crucial in stock forecasting. To handle longer sequence data, new strategies and network architectures are continuously proposed to enable models to learn more time-step information and deeply mine market rules. Since the introduction of the RNN method, research in time series analysis has rapidly increased, followed by the improved LSTM and GRU models that further address the problem of vanishing gradients encountered by RNNs in processing long sequences. Additionally, the introduction of the Transformer enhanced the model's parallel processing capabilities and long-distance dependency-capturing abilities, but traditional attention mechanisms are inefficient and costly for ultra-long sequences. To this end, scholars have proposed the Informer model using sparse attention to reduce computational costs. Meanwhile, TCNs designed based on convolutional neural networks ensure the unidirectionality of information flow through causal convolutions, further enhancing the model's parallel processing capabilities.

Recently, structured state space models (SSMs) [154,155] have been proposed in the field of time series analysis. Inspired by classic state space models [156], SSMs can be computed efficiently through recursive or convolutional methods, and they handle sequence lengths with linear or near-linear scalability. It means that SSMs can process longer sequence data without significantly increasing computational costs, making them very suitable for time series prediction tasks. However, their application in stock prediction is limited because SSMs capture long-range dependencies based solely on distance, lacking effective mechanisms to identify and differentiate the importance of information. Albert Gu and Tri Dao addressed this issue in their research [157], developing a selective state space model called Mamba. Based on a selective mechanism, this model can differentiate between critical and non-critical information like a Transformer while maintaining the linear scalability of SSMs, enabling it to outperform Transformers in handling ultra-long sequence data. Experimental results show that the Mamba model demonstrates strong potential in processing sequence and image data. Shi et al. [158] applied the Mamba model to stock price prediction, introducing MambaStock. The results indicate that Mamba can effectively use historical stock market data to predict future stock prices without complex feature engineering or preprocessing.

Continual Learning [159], also known as incremental learning or lifelong learning, is a method designed to allow machine learning models to continuously absorb new information and adapt to new situations while retaining knowledge of old data, with the core challenge being to solve "catastrophic forgetting". This learning mode is particularly suited to dynamic environments where new data continuously emerges, and tasks may change over time. In stock prediction, data from the stock market is constantly generated over time, and existing prediction models are primarily based on fixed-length datasets. Although this method can learn historical market trends, it cannot learn market changes in real time. Continual learning methods can learn and adapt to new information while retaining previously learned knowledge.

In summary, effectively modelling long-term dependencies is a core challenge in time series analysis, requiring models to capture long-sequence information and identify key dependencies. In recent years, structured state-space models like Mamba, through introducing selective mechanisms, have achieved efficient processing of ultra-long sequences, showing great potential. Meanwhile, the continual learning paradigm provides a new solution for long-term dependency issues in dynamically changing environments, focusing on preventing "catastrophic forgetting" and continuously learning new knowledge. In the future, fully utilizing these advanced models and learning paradigms will help capture and utilize long-term historical dependencies more accurately, thereby enhancing the overall performance of time series modelling.

5.3. Lack of universality and interpretability

Universality and interpretability are critical issues in the current field of deep learning. Most existing neural network-based stock forecasting models can only predict limited datasets in specific environments. However, as economic globalization deepens, financial markets have become highly intertwined entities with complex interconnections among countries, markets, and stocks. Therefore, developing models with broad applicability and high interpretability to adapt to various market conditions and articulate prediction logic has become an urgent need, and future research is focused on the field of stock forecasting.

Neural network models are often viewed as black-box systems; thus, their outputs generally lack interpretability [160]. In stock prediction, which involves significant financial decisions, the transparency and interpretability of models directly impact their effectiveness and credibility. Interpretability learning, which reveals the intrinsic logic of the model's decision-making process, is essential for complying with regulatory requirements, boosting investor confidence, timely strategy adjustments, and optimizing portfolio management. To enhance the local interpretability of prediction models, methods such as LIME, SHAP, or EBM Boosting can be used [161], while global interpretability can be improved using SHAP algorithms or XGBoost techniques [162]. Integrating these methods effectively with prediction models is vital to future research to enhance the interpretability of stock predictions.

The year 2023 has been dubbed the inaugural year of large language models (LLMs) [163], with explosive growth in general significant model technologies prompting various industries and research fields to explore integration with large language models actively. These models' applications have expanded beyond essential dialogues to include education, healthcare, law, and more [164–166]. Specifically, using large language models for stock prediction can be categorized into three methods: direct querying of LLMs, LLM fine-tuning through custom design, and integrating LLMs as a feature enhancement tool into time series models [167]. The first two methods involve using financial market data to specifically train or fine-tune the model to produce accurate predictions [133]. The third method is based on extracting features from a frozen large language model, which downstream models then use to learn and output predictions, such as using LLMs to analyse financial textual data to extract sentiment indicators for market trend prediction [131,132]. Based on the above discussion, we believe that stock prediction methods based on LLMs will have significant future research potential and development space.

5.4. Inconsistency in evaluation metrics and datasets

The majority of studies reviewed in this article use different datasets and evaluation metrics, which hinders the comparison and learning among various methods in the field of stock forecasting to some extent. The primary goal of stock forecasting is to forecast future stock price movements based on market prices and trends, providing decision support for investors and analysts, thereby profiting in natural market conditions. However, most current stock forecasting methods focus only on intermediate metrics in experimental data, such as regression and classification indicators, while neglecting key financial evaluation metrics that reflect the model's ability to help achieve profitability, such as Return Ratio (RR) and Sharpe Ratio (SR). Additionally, the lack of a unified benchmark dataset means that methods often perform well only on specific datasets and may experience a sharp decline in performance when faced with data from different markets. Therefore, future stock forecasting research should focus on the performance of financial evaluation metrics and concentrate on its fundamental goal—to better help investors manage risk, optimize investment strategies, and improve the efficiency and effectiveness of economic decisions. We should promote the establishment of a unified stock forecasting benchmark dataset and a clear evaluation system to facilitate research interaction and progress within the field.

6. Conclusion

This review comprehensively examines data-driven stock forecasting models based on neural networks, analysing 63 journal and conference articles from 2015 to 2023. Specifically, we categorized the methods according to their network architecture, covering stock forecasting approaches based on RNNs, CNNs, Transformers, GNNs, GANs, and LLMs, and discussed the functionalities, advantages, and classic variants of these network structures. These models demonstrate that, compared to traditional stock forecasting models, deep learning methods represented by neural networks show outstanding performance, enabling even non-professionals to train prediction models that surpass those developed with years of professional experience using historical financial data. Additionally, we organized commonly used datasets and evaluation metrics in the field of stock forecasting, detailed their calculation methods, and summarized them in table format.

We also discussed key issues such as data representation, challenges in modelling long-term dependencies, the need for models to achieve generality and interpretability, and the lack of uniformity in evaluation standards and datasets. These are not only persistent challenges in the field of stock forecasting but also fertile grounds for future research, pointing the way towards building more robust, more interpretable, and more effective prediction models. Overall, as the financial industry continues to develop rapidly, driven by technological advances and increasing data availability, the role of neural networks in stock forecasting is expected to expand further. Continuous innovation and research are essential to address these challenges and fully exploit the potential of neural network models in this dynamic field. We hope this review provides a solid foundation for researchers and practitioners, aiding them in achieving further success as they use neural networks to advance the field of financial prediction.

CRediT authorship contribution statement

Wuzhida Bao: Writing – original draft, Validation, Software, Methodology. **Yuting Cao:** Validation, Investigation, Formal analysis. **Yin Yang:** Supervision, Project administration, Methodology, Investigation. **Hangjun Che:** Methodology, Investigation, Conceptualization. **Junjian Huang:** Visualization, Resources, Formal analysis. **Shiping Wen:** Writing – review & editing, Supervision, Resources.

Declaration of competing interest

There is no conflict of interest in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgement

The statements made herein are solely the responsibility of the author[s].

References

[1] C. Steindel, S.C. Ludvigson, How important is the stock market effect on consumption? Econ. Policy Rev. 5 (2) (1999).
[2] L. Bachelier, Théorie de la spéculation, in: Annales Scientifiques De L' École Normale Supérieure, Vol. 17, 1900, pp. 21–86.
[3] P.H. Cootner, The random character of stock market prices, 1964, (No Title).
[4] E.F. Fama, The behavior of stock-market prices, J. Bus. 38 (1) (1965) 34–105.
[5] E.F. Fama, Efficient capital markets, J. Finance 25 (2) (1970) 383–417.
[6] A.W. Lo, A.C. MacKinlay, Stock market prices do not follow random walks: Evidence from a simple specification test, Rev. Final. Stud. 1 (1) (1988) 41–66.
[7] R.J. Shiller, Do stock prices move too much to be justified by subsequent changes in dividends? Am. Econ. Rev. 71 (3) (1981) 421–436.

[8] E.F. Fama, L. Fisher, M.C. Jensen, R. Roll, The adjustment of stock prices to new information, Int. Econ. Rev. 10 (1) (1969) 1–21.
[9] E.H. Kim, A mean–variance theory of optimal capital structure and corporate debt capacity, J. Finance 33 (1) (1978) 45–63.
[10] D. Bhuriya, G. Kaushal, A. Sharma, U. Singh, Stock market predication using a linear regression, in: 2017 International Conference of Electronics, Communication and Aerospace Technology, ICECA, Vol. 2, IEEE, 2017, pp. 510–513.
[11] J.F. De Oliveira, T.B. Ludermir, A hybrid evolutionary decomposition system for time series forecasting, Neurocomputing 180 (2016) 27–34.
[12] R. Lewis, G.C. Reinsel, Prediction of multivariate time series by autoregressive model fitting, J. Multivariate Anal. 16 (3) (1985) 393–411.
[13] R. Haining, The moving average model for spatial interaction, Trans. Inst. Br. Geogr. (1978) 202–225.
[14] A.A. Ariyo, A.O. Adewumi, C.K. Ayo, Stock price prediction using the arima model, in: 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, IEEE, 2014, pp. 106–112.
[15] M. Kumar, M. Thenmozhi, Forecasting stock index returns using arimasvm, arima-ann, and arima-random forest hybrid models, Int. J. Bank. Account. Finance 5 (3) (2014) 284–308.
[16] Y. Wang, Y. Guo, Forecasting method of stock market volatility in time series data based on mixed model of arima and xgboost, China Commun. 17 (3) (2020) 205–221.
[17] M.R. Hassan, B. Nath, M. Kirley, A fusion model of hmm, ann and ga for stock market forecasting, Expert Syst. Appl. 33 (1) (2007) 171–180.
[18] Y. Kara, M.A. Boyacioglu, Ö.K. Baykan, Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange, Expert Syst. Appl. 38 (5) (2011) 5311–5319.
[19] K.-j. Kim, Financial time series forecasting using support vector machines, Neurocomputing 55 (1–2) (2003) 307–319.
[20] L. Wang, H. Zou, J. Su, L. Li, S. Chaudhry, An arima-ann hybrid model for time series forecasting, Syst. Res. Behav. Sci. 30 (3) (2013) 244–259.
[21] Q. Wen, Z. Yang, Y. Song, P. Jia, Automatic stock decision support system based on box theory and svm algorithm, Expert Syst. Appl. 37 (2) (2010) 1015–1022.
[22] R.K. Nayak, D. Mishra, A.K. Rath, A naïve svm-knn based stock market trend reversal analysis for indian benchmark indices, Appl. Soft Comput. 35 (2015) 670–680.
[23] A. Booth, E. Gerding, F. McGroarty, Automated trading with performance weighted random forests and seasonality, Expert Syst. Appl. 41 (8) (2014) 3651–3661.
[24] S. Basak, S. Kar, S. Saha, L. Khaidem, S.R. Dey, Predicting the direction of stock market prices using tree-based classifiers, North Am. J. Econ. Finance 47 (2019) 552–567.
[25] G.S. Atsalakis, K.P. Valavanis, Surveying stock market forecasting techniques–part ii: Soft computing methods, Expert Syst. Appl. 36 (3) (2009) 5932–5941.
[26] Y. Li, W. Ma, Applications of artificial neural networks in financial economics: a survey, in: 2010 International Symposium on Computational Intelligence and Design, Vol. 1, IEEE, 2010, pp. 211–214.
[27] S. Soni, Applications of anns in stock market prediction: a survey, Int. J. Comput. Sci. Eng. Technol. 2 (3) (2011) 71–83.
[28] M. Tkáč, R. Verner, Artificial neural networks in business: Two decades of research, Appl. Soft Comput. 38 (2016) 788–804.
[29] D.P. Gandhmal, K. Kumar, Systematic analysis and review of stock market prediction techniques, Comp. Sci. Rev. 34 (2019) 100190.
[30] A.K. Nassirtoussi, S. Aghabozorgi, T.Y. Wah, D.C.L. Ngo, Text mining for market prediction: A systematic review, Expert Syst. Appl. 41 (16) (2014) 7653–7670.
[31] Q. Li, Y. Chen, J. Wang, Y. Chen, H. Chen, Web media and stock markets: A survey and future directions from a big data perspective, IEEE Trans. Knowl. Data Eng. 30 (2) (2017) 381–399.
[32] F.Z. Xing, E. Cambria, R.E. Welsch, Natural language based financial forecasting: a survey, Artif. Intell. Rev. 50 (1) (2018) 49–73.
[33] T.B. Shahi, A. Shrestha, A. Neupane, W. Guo, Stock price forecasting with deep learning: A comparative study, Mathematics 8 (9) (2020) 1441.
[34] O.B. Sezer, M.U. Gudelek, A.M. Ozbayoglu, Financial time series forecasting with deep learning: A systematic literature review: 2005–2019, Appl. Soft Comput. 90 (2020) 106181.
[35] W. Jiang, Applications of deep learning in stock market prediction: recent progress, Expert Syst. Appl. 184 (2021) 115537.
[36] F. Eckerli, J. Osterrieder, Generative adversarial networks in finance: an overview, 2021, arXiv preprint arXiv:2106.06364.
[37] M.M. Kumbure, C. Lohrmann, P. Luukka, J. Porras, Machine learning techniques and data for stock market forecasting: A literature review, Expert Syst. Appl. 197 (2022) 116659.
[38] J. Zou, Q. Zhao, Y. Jiao, H. Cao, Y. Liu, Q. Yan, E. Abbasnejad, L. Liu, J.Q. Shi, Stock market prediction via deep learning techniques: A survey, 2022, arXiv preprint arXiv:2212.12717.
[39] R.P. Masini, M.C. Medeiros, E.F. Mendes, Machine learning advances for time series forecasting, J. Econ. Surv. 37 (1) (2023) 76–111.
[40] A.M. Rather, A. Agarwal, V. Sastry, Recurrent neural network and a hybrid model for prediction of stock returns, Expert Syst. Appl. 42 (6) (2015) 3234–3241.
[41] Z. Berradi, M. Lazaar, Integration of principal component analysis and recurrent neural network to forecast the stock price of casablanca stock exchange, Procedia Comput. Sci. 148 (2019) 55–61.
[42] L. Yunpeng, H. Di, B. Junpeng, Q. Yong, Multi-step ahead time series forecasting for different data patterns based on lstm recurrent neural network, in: 2017 14th Web Information Systems and Applications Conference, WISA, IEEE, 2017, pp. 305–310.
[43] S. Liu, G. Liao, Y. Ding, Stock transaction prediction modeling and analysis based on lstm, in: 2018 13th IEEE Conference on Industrial Electronics and Applications, ICIEA, IEEE, 2018, pp. 2787–2790.
[44] A.Q. Md, S. Kapoor, C.J. Av, A.K. Sivaraman, K.F. Tee, H. Sabireen, N. Janakiraman, Novel optimization approach for stock price forecasting using multi-layered sequential lstm, Appl. Soft Comput. 134 (2023) 109830.
[45] Y. Baek, H.Y. Kim, Modaugnet: A new forecasting framework for stock market index value with an overfitting prevention lstm module and a prediction lstm module, Expert Syst. Appl. 113 (2018) 457–480.
[46] H.Y. Kim, C.H. Won, Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models, Expert Syst. Appl. 103 (2018) 25–37.
[47] K. Kumar, M.T.U. Haider, Enhanced prediction of intra-day stock market using metaheuristic optimization on rnn–lstm network, New Gener. Comput. 39 (1) (2021) 231–272.
[48] Y. Yujun, Y. Yimei, X. Jianhua, A hybrid prediction method for stock price using lstm and ensemble emd, Complexity 2020 (2020) 1–16.
[49] Y. Zhang, B. Yan, M. Aasma, A novel deep learning framework: Prediction and analysis of financial time series using ceemd and lstm, Expert Syst. Appl. 159 (2020) 113609.
[50] Y. Lin, Z. Lin, Y. Liao, Y. Li, J. Xu, Y. Yan, Forecasting the realized volatility of stock price index: A hybrid model integrating ceemdan and lstm, Expert Syst. Appl. 206 (2022) 117736.
[51] M. Yang, J. Wang, Adaptability of financial time series prediction based on bilstm, Procedia Comput. Sci. 199 (2022) 18–25.
[52] J. Vaziri, D. Farid, M. Nazemi Ardakani, S.M. Hosseini Bamakan, M. Shahlaei, A time-varying stock portfolio selection model based on optimized pso-bilstm and multi-objective mathematical programming under budget constraints, Neural Comput. Appl. 35 (25) (2023) 18445–18470.
[53] D.L. Minh, A. Sadeghi-Niaraki, H.D. Huy, K. Min, H. Moon, Deep learning approach for short-term stock trends prediction based on twostream gated recurrent unit network, Ieee Access 6 (2018) 55392–55404.
[54] J. Li, J. Wang, Forcasting of energy futures market and synchronization based on stochastic gated recurrent unit model, Energy 213 (2020) 118787.
[55] U. Gupta, V. Bhattacharjee, P.S. Bishnu, Stocknet—gru based stock index prediction, Expert Syst. Appl. 207 (2022) 117986.
[56] L.R. Medsker, L. Jain, Recurrent neural networks, Des. Appl. 5 (64–67) (2001) 2.
[57] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, Lstm: A search space odyssey, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2016) 2222–2232.
[58] G. Rilling, P. Flandrin, P. Goncalves, On empirical mode decomposition and its algorithms, in: IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing, IEEE, Grado, 2003, pp. 8–11.
[59] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (11) (1997) 2673–2681.
[60] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder–decoder for statistical machine translation, 2014, arXiv preprint arXiv:1406.1078.
[61] S. Yang, X. Yu, Y. Zhou, Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example, in: 2020 International Workshop on Electronic Communication and Artificial Intelligence, IWECAI, IEEE, 2020, pp. 98–101.
[62] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V.K. Menon, K. Soman, Stock price prediction using lstm, rnn and cnn-sliding window model, in: 2017 International Conference on Advances in Computing, Communications and Informatics, icacci, IEEE, 2017, pp. 1643–1647.
[63] E. Hoseinzade, S. Haratizadeh, Cnnpred: Cnn-based stock market prediction using a diverse set of variables, Expert Syst. Appl. 129 (2019) 273–285.
[64] D.M. Durairaj, B.K. Mohan, A convolutional neural network based approach to financial time series prediction, Neural Comput. Appl. 34 (16) (2022) 13319–13337.
[65] L. Börjesson, M. Singull, Forecasting financial time series through causal and dilated convolutional neural networks, Entropy 22 (10) (2020) 1094.
[66] Y. Zhao, M. Khushi, Wavelet denoised-resnet cnn and lightgbm method to predict forex rate of change, in: 2020 International Conference on Data Mining Workshops, ICDMW, IEEE, 2020, pp. 385–391.
[67] M. Liu, M. Huang, Y. Zhang, W. Feng, J. Lai, X. Li, Using deep residual networks to deal with financial risk control problems, in: Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence, 2018, pp. 1–6.

[68] P. Khodaee, A. Esfahanipour, H.M. Taheri, Forecasting turning points in stock price by applying a novel hybrid cnn-lstm-resnet model fed by 2d segmented images, Eng. Appl. Artif. Intell. 116 (2022) 105464.
[69] Y. Liu, H. Dong, X. Wang, S. Han, Time series prediction based on temporal convolutional network, in: 2019 IEEE/ACIS 18th International Conference on Computer and Information Science, ICIS, IEEE, 2019, pp. 300–305.
[70] S. Deng, N. Zhang, W. Zhang, J. Chen, J.Z. Pan, H. Chen, Knowledgedriven stock trend prediction and explanation via temporal convolutional network, in: Companion Proceedings of the 2019 World Wide Web Conference, 2019, pp. 678–685.
[71] Y. Yao, Z.-y. Zhang, Y. Zhao, Stock index forecasting based on multivariate empirical mode decomposition and temporal convolutional networks, Appl. Soft Comput. 142 (2023) 110356.
[72] A. Kanwal, M.F. Lau, S.P. Ng, K.Y. Sim, S. Chandrasekaran, Bicudnnlstm-1dcnn—a hybrid deep learning-based predictive model for stock price prediction, Expert Syst. Appl. 202 (2022) 117123.
[73] J.M.-T. Wu, Z. Li, N. Herencsar, B. Vo, J.C.-W. Lin, A graph-based cnn-lstm stock price prediction algorithm with leading indicators, Multimedia Syst. 29 (3) (2023) 1751–1770.
[74] I.E. Livieris, E. Pintelas, P. Pintelas, A cnn–lstm model for gold price time-series forecasting, Neural Comput. Appl. 32 (2020) 17351–17360.
[75] I.E. Livieris, N. Kiriakidou, S. Stavroyiannis, P. Pintelas, An advanced cnn-lstm model for cryptocurrency forecasting, Electronics 10 (3) (2021) 287.
[76] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, Recent advances in convolutional neural networks, Pattern Recognit. 77 (2018) 354–377.
[77] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012).
[78] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
[79] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018, arXiv preprint arXiv:1803.01271.
[80] L. Yang, T.L.J. Ng, B. Smyth, R. Dong, Html: Hierarchical transformer-based multi-task learning for volatility prediction, in: Proceedings of the Web Conference 2020, 2020, pp. 441–451.
[81] E. Ramos-Pérez, P.J. Alonso-González, J.J. Núñez-Velázquez, Multitransformer: A new neural network-based architecture for forecasting s & p volatility, Mathematics 9 (15) (2021) 1794.
[82] C. Wang, Y. Chen, S. Zhang, Q. Zhang, Stock market index prediction using deep transformer model, Expert Syst. Appl. 208 (2022) 118128.
[83] B. Ma, Y. Wang, Y. Lu, T. Hu, J. Xu, P. Houlihan, Stockformer: A swing trading strategy based on stl decomposition and self-attention networks, 2023, arXiv preprint arXiv:2401.06139.
[84] F. Mercier, M. Narsimhan, Discovering material information using hierarchical reformer model on financial regulatory filings, 2022, arXiv preprint arXiv:2204.05979.
[85] Y. Lu, H. Zhang, Q. Guo, Stock and market index prediction using informer network, 2023, arXiv preprint arXiv:2305.14382.
[86] S. Ren, X. Wang, X. Zhou, Y. Zhou, A novel hybrid model for stock price forecasting integrating encoder forest and informer, Expert Syst. Appl. 234 (2023) 121080.
[87] H. Liu, D. Chen, W. Wei, Z. Wei, Long-term stock price forecast based on pso-informer model, in: Fifth International Conference on Computer Information Science and Artificial Intelligence, CISAI 2022, Vol. 12566, SPIE, 2023, pp. 288–294.
[88] C. Liu, K. Ruan, X. Ma, Dmeformer: A newly designed dynamic model ensemble transformer for crude oil futures prediction, Heliyon 9 (6) (2023).
[89] Z. Niu, G. Zhong, H. Yu, A review on the attention mechanism of deep learning, Neurocomputing 452 (2021) 48–62.
[90] P. Shaw, J. Uszkoreit, A. Vaswani, Self-attention with relative position representations, 2018, arXiv preprint arXiv:1803.02155.
[91] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[92] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, I. Titov, Analyzing multihead self-attention: Specialized heads do the heavy lifting, the rest can be pruned, 2019, arXiv preprint arXiv:1905.09418.
[93] N. Kitaev, Ł. Kaiser, A. Levskaya, Reformer: The efficient transformer, 2020, arXiv preprint arXiv:2001.04451.
[94] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 11106–11115.
[95] X. Zhang, F. Wei, M. Zhou, Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization, 2019, arXiv preprint arXiv:1905.06566.
[96] A. Zeng, M. Chen, L. Zhang, Q. Xu, Are transformers effective for time series forecasting? in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023, pp. 11121–11128.
[97] Y. Yang, Z. Wei, Q. Chen, L. Wu, Using external knowledge for financial event prediction based on graph neural networks, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 2161–2164.
[98] D. Matsunaga, T. Suzumura, T. Takahashi, Exploring graph neural networks for stock market predictions with rolling window analysis, 2019, arXiv preprint arXiv:1909.10660.
[99] R. Sawhney, S. Agarwal, A. Wadhwa, T. Derr, R.R. Shah, Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 497–504.
[100] D. Cheng, F. Yang, S. Xiang, J. Liu, Financial time series forecasting with multi-modality graph neural network, Pattern Recognit. 121 (2022) 108218.
[101] Y. Zhao, H. Du, Y. Liu, S. Wei, X. Chen, F. Zhuang, Q. Li, G. Kou, Stock movement prediction based on bi-typed hybrid-relational market knowledge graph via dual attention networks, IEEE Trans. Knowl. Data Eng. (2022).
[102] Y. Chen, Z. Wei, X. Huang, Incorporating corporation relationship via graph convolutional neural networks for stock price prediction, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 1655–1658.
[103] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, T.-S. Chua, Temporal relational ranking for stock prediction, ACM Trans. Inf. Syst. (TOIS) 37 (2) (2019) 1–30.
[104] J. Gao, X. Ying, C. Xu, J. Wang, S. Zhang, Z. Li, Graph-based stock recommendation by time-aware relational attention network, ACM Trans. Knowl. Discov. Data (TKDD) 16 (1) (2021) 1–21.
[105] W. Chen, M. Jiang, W.-G. Zhang, Z. Chen, A novel graph convolutional feature based convolutional neural network for stock trend prediction, Inform. Sci. 556 (2021) 67–94.
[106] S. Li, J. Wu, X. Jiang, K. Xu, Chart gcn: Learning chart information with a graph convolutional network for stock movement prediction, Knowl.-Based Syst. 248 (2022) 108842.
[107] X. Ma, T. Zhao, Q. Guo, X. Li, C. Zhang, Fuzzy hypergraph network for recommending top-k profitable stocks, Inform. Sci. 613 (2022) 239–255.
[108] G. Song, T. Zhao, S. Wang, H. Wang, X. Li, Stock ranking prediction using a graph aggregation network based on stock price and stock relationship information, Inform. Sci. 643 (2023) 119236.
[109] H. Peng, K. Dong, J. Yang, Stock price movement prediction based on relation type guided graph convolutional network, Eng. Appl. Artif. Intell. 126 (2023) 106948.
[110] R. Sawhney, S. Agarwal, A. Wadhwa, R. Shah, Deep attentive learning for stock movement prediction from social media text and company correlations, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2020, pp. 8415–8426.
[111] Y.-L. Hsu, Y.-C. Tsai, C.-T. Li, Fingat: Financial graph attention networks for recommending top-kk profitable stocks, IEEE Trans. Knowl. Data Eng. 35 (1) (2023) 469–481, http://dx.doi.org/10.1109/TKDE.2021.3079496.
[112] S. Feng, C. Xu, Y. Zuo, G. Chen, F. Lin, J. XiaHou, Relation-aware dynamic attributed graph attention network for stocks recommendation, Pattern Recognit. 121 (2022) 108119.
[113] X. Ma, X. Li, W. Feng, L. Fang, C. Zhang, Dynamic graph construction via motif detection for stock prediction, Inf. Process. Manage. 60 (6) (2023) 103480.
[114] Z. Lei, C. Zhang, Y. Xu, X. Li, Dr-gat: Dynamic routing graph attention network for stock recommendation, Inform. Sci. 654 (2024) 119833.
[115] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S.Y. Philip, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. 32 (1) (2020) 4–24.
[116] S. Zhang, H. Tong, J. Xu, R. Maciejewski, Graph convolutional networks: a comprehensive review, Comput. Soc. Netw. 6 (1) (2019) 1–23.
[117] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, 2017, arXiv preprint arXiv:1710.10903.
[118] X. Zhou, Z. Pan, G. Hu, S. Tang, C. Zhao, Stock market prediction on high-frequency data using generative adversarial nets, Math. Probl. Eng. (2018).
[119] G. Mariani, Y. Zhu, J. Li, F. Scheidegger, R. Istrate, C. Bekas, A.C.I. Malossi, Pagan: Portfolio analysis with generative adversarial networks, 2019, arXiv preprint arXiv:1909.10578.
[120] K. Zhang, G. Zhong, J. Dong, S. Wang, Y. Wang, Stock market prediction based on generative adversarial network, Procedia Comput. Sci. 147 (2019) 400–406.
[121] J.-L. Wu, X.-R. Tang, C.-H. Hsu, A prediction model of stock market trading actions using generative adversarial network and piecewise linear representation approaches, Soft Comput. 27 (12) (2023) 8209–8222.
[122] R. Jadhav, S. Sinha, S. Wattamwar, P. Kosamkar, Leveraging market sentiment for stock price prediction using gan, in: 2021 2nd Global Conference for Advancement in Technology, GCAT, IEEE, 2021, pp. 1–6.
[123] P. Muthukumar, J. Zhong, A stochastic time series model for predicting financial trends using nlp, 2021, arXiv preprint arXiv:2102.01290.
[124] A. Koshiyama, N. Firoozye, P. Treleaven, Generative adversarial networks for financial trading strategies fine-tuning and combination, Quant. Finance 21 (5) (2021) 797–813.
[125] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, 2014, arXiv:1406.2661.

[126] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A.A. Bharath, Generative adversarial networks: An overview, IEEE Signal Process. Mag. 35 (1) (2018) 53–65.
[127] M. Mirza, S. Osindero, Conditional generative adversarial nets, 2014, arXiv preprint arXiv:1411.1784.
[128] F. Zhao, X. Li, Y. Gao, Y. Li, Z. Feng, C. Zhang, Multi-layer features ablation of bert model and its application in stock trend prediction, Expert Syst. Appl. 207 (2022) 117958.
[129] M.G. Sousa, K. Sakiyama, L. de Souza Rodrigues, P.H. Moraes, E.R. Fernandes, E.T. Matsubara, Bert for stock market sentiment analysis, in: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence, ICTAI, IEEE, 2019, pp. 1597–1601.
[130] T. Sidogi, R. Mbuvha, T. Marwala, Stock price prediction using sentiment analysis, in: 2021 IEEE International Conference on Systems, Man, and Cybernetics, SMC, IEEE, 2021, pp. 46–51.
[131] J.Z.G. Hiew, X. Huang, H. Mou, D. Li, Q. Wu, Y. Xu, Bert-based financial sentiment index and lstm-based stock return predictability, 2022, arXiv:1906.09024.
[132] A.H. Huang, H. Wang, Y. Yang, Finbert: A large language model for extracting information from financial text, Contemp. Account. Res. 40 (2) (2023) 806–841.
[133] A. Garza, M. Mergenthaler-Canseco, Timegpt-1, 2023, arXiv preprint arXiv:2310.03589.
[134] M. Jin, S. Wang, L. Ma, Z. Chu, J.Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, et al., Time-llm: Time series forecasting by reprogramming large language models, 2023, arXiv preprint arXiv:2310.01728.
[135] X. Yu, Z. Chen, Y. Ling, S. Dong, Z. Liu, Y. Lu, Temporal data meets llm–explainable financial time series forecasting, 2023, arXiv preprint arXiv:2306.11025.
[136] W.X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, A survey of large language models, 2023, arXiv preprint arXiv:2303.18223.
[137] M. Shanahan, Talking about large language models, Commun. ACM 67 (2) (2024) 68–79.
[138] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.
[139] D. Araci, Finbert: Financial sentiment analysis with pre-trained language models, 2019, arXiv preprint arXiv:1908.10063.
[140] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training, 2018, URL https://api.semanticscholar.org/CorpusID:49313245.
[141] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, 2023, arXiv preprint arXiv:2302.13971.
[142] X. Chen, S. Jia, Y. Xiang, A review: Knowledge reasoning over knowledge graph, Expert Syst. Appl. 141 (2020) 112948.
[143] J.D. Hamilton, Time Series Analysis, Princeton University Press, 2020.
[144] J. Sun, Y. Qing, C. Liu, J. Lin, Self-fts: A self-supervised learning 25 method for financial time series representation in stock intraday trading, in: 2022 IEEE 20th International Conference on Industrial Informatics, INDIN, IEEE, 2022, pp. 501–506.
[145] Y. Ma, C. Ventre, M. Polukarov, Denoised labels for financial time series data via self-supervised learning, in: Proceedings of the Third ACM International Conference on AI in Finance, 2022, pp. 471–479.
[146] Z. Ying, D. Cheng, C. Chen, X. Li, P. Zhu, Y. Luo, Y. Liang, Predicting stock market trends with self-supervised learning, Neurocomputing 568 (2024) 127033.
[147] K. Zhang, Q. Wen, C. Zhang, R. Cai, M. Jin, Y. Liu, J.Y. Zhang, Y. Liang, G. Pang, D. Song, Self-supervised learning for time series analysis: Taxonomy, progress, and prospects, IEEE Trans. Pattern Anal. Mach. Intell. (2024).
[148] Z. Wang, T. Oates, Imaging time-series to improve classification and imputation, 2015, arXiv preprint arXiv:1506.00327.
[149] J.-F. Chen, W.-L. Chen, C.-P. Huang, S.-H. Huang, A.-P. Chen, Financial time-series data analysis using deep convolutional neural networks, in: 2016 7th International Conference on Cloud Computing and Big Data, CCBD, IEEE, 2016, pp. 87–92.
[150] S. Barra, S.M. Carta, A. Corriga, A.S. Podda, D.R. Recupero, Deep learning and time series-to-image encoding for financial forecasting, IEEE/CAA J. Autom. Sin. 7 (3) (2020) 683–692.
[151] A. Ghasemieh, R. Kashef, An enhanced wasserstein generative adversarial network with gramian angular fields for efficient stock market prediction during market crash periods, Appl. Intell. 53 (23) (2023) 28479–28500.
[152] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A.Y. Ng, Multimodal deep learning, in: Proceedings of the 28th International Conference on Machine Learning, ICML-11, 2011, pp. 689–696.
[153] R. Liu, H. Liu, H. Huang, B. Song, Q. Wu, Multimodal multiscale dynamic graph convolution networks for stock price prediction, Pattern Recognit. 149 (2024) 110211.
[154] A. Gu, I. Johnson, K. Goel, K. Saab, T. Dao, A. Rudra, C. Ré, Combining recurrent, convolutional,and continuous-time models with linear state space layers, Adv. Neural Inf. Process. Syst. 34 (2021) 572–585.
[155] A. Gu, K. Goel, C. Ré, Efficiently modeling long sequences with structured state spaces, 2021, arXiv preprint arXiv:2111.00396.
[156] R.E. Kalman, A new approach to linear filtering and prediction problems, 1960.
[157] A. Gu, T. Dao, Mamba: Linear-time sequence modeling with selective state spaces, 2023, arXiv preprint arXiv:2312.00752.
[158] Z. Shi, Mambastock: Selective state space model for stock prediction, 2024, arXiv preprint arXiv:2402.18959.
[159] L. Wang, X. Zhang, H. Su, J. Zhu, A comprehensive survey of continual learning: Theory, method and application, IEEE Trans. Pattern Anal. Mach. Intell. (2024).
[160] L.H. Gilpin, D. Bau, B.Z. Yuan, A. Bajwa, M. Specter, L. Kagal, Explaining explanations: An overview of interpretability of machine learning, in: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics, DSAA, IEEE, 2018, pp. 80–89.
[161] Z. Zhang, H. Al Hamadi, E. Damiani, C.Y. Yeun, F. Taher, Explainable artificial intelligence applications in cyber security: State-of-the-art in research, IEEE Access 10 (2022) 93104–93139.
[162] Z. Li, Extracting spatial effects from machine learning model using local interpretation method: An example of shap and xgboost, Comput. Environ. Urban Syst. 96 (2022) 101845.
[163] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, A survey on evaluation of large language models, ACM Trans. Intell. Syst. Technol. 15 (3) (2024) 1–45.
[164] L. Yan, L. Sha, L. Zhao, Y. Li, R. Martinez-Maldonado, G. Chen, X. Li, Y. Jin, D. Gašević, Practical and ethical challenges of large language models in education: A systematic scoping review, Br. J. Educ. Technol. 55 (1) (2024) 90–112.
[165] A.J. Thirunavukarasu, D.S.J. Ting, K. Elangovan, L. Gutierrez, T.F. Tan, D.S.W. Ting, Large language models in medicine, Nat. Med. 29 (8) (2023) 1930–1940.
[166] X. Yang, Z. Wang, Q. Wang, K. Wei, K. Zhang, J. Shi, Large language models for automated q & a involving legal documents: a survey on algorithms, frameworks and applications, Int. J. Web Inf. Syst. (2024).
[167] Y. Jiang, Z. Pan, X. Zhang, S. Garg, A. Schneider, Y. Nevmyvaka, D. Song, Empowering time series analysis with large language models: A survey, 2024, arXiv preprint arXiv:2402.03182.