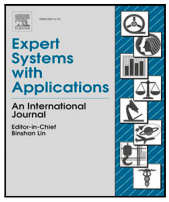
Expert Systems With Applications 250 (2024) 123948

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Enhanced stock price forecasting through a regularized ensemble framework with graph convolutional networks

Dongbo Ma, Da Yuan *

School of Computer Science and Technology, Shandong Technology and Business University, Yantai 264005, China

ARTICLE INFO

Keywords:
Stock price prediction
Variational mode decomposition
Ensemble graph convolutional network
Regularization

ABSTRACT

Stock prediction is a prominent and challenging topic of interest for both investors and researchers. With the introduction of Graph Convolutional Networks (GCNs) to financial data analysis, many researchers have attempted to use GCNs to exploit the interrelationships among stock data. However, a notable research gap remains in the application of GCNs to establish associations among transaction data, which inherently exhibit more stable and well-defined correlations. Hence, this study proposes a regularized ensemble framework with graph convolutional networks for stock price prediction. The framework employs multiple multi-graph convolutional networks to construct an ensemble graph convolutional network, elucidating spatiotemporal relationships within transaction data across various time scales. Additionally, the Variational Modal Decomposition (VMD) method is utilized to extract features from each dimensional sequence of the transaction data, enabling the capture of features across different time scales and mitigating non-stationarity and noise in the data. Furthermore, a regularization term incorporating trend accuracy is introduced to strike a balance between numerical precision and trend accuracy, thereby enhancing the overall model performance. Finally, this study presents a specialized investment strategy designed for stock price prediction. Our approach yields substantial improvements on SSE and DJIA datasets, with ACC and R^2 improving by over 15% and 10%, respectively, accompanied by reductions of more than 40% in RMSE, MAE, and MAPE compared to the baseline. These results underscore the effectiveness, robustness, and adaptability of our proposed methodology. Our codes and data can be accessed at https://github.com/WonderNothing/REGCN/.

1. Introduction

Financial time series prediction is widely acknowledged as a significant challenge in both time series analysis and machine learning literature (Tay & Cao, 2001). The accurate prediction of stock prices is paramount for making informed investment decisions and effectively participating in capital markets. While the predictability of stock markets remains a subject of ongoing debate, numerous studies have confirmed that stocks can be predicted to a certain extent (Bollerslev et al., 2014).

The landscape of stock prediction models primarily consists of two categories: statistical models and Artificial Intelligence (AI) models. AI-based models outperform their statistical counterparts in managing non-linearity, dynamics, instability, and noisy stock price data, thereby achieving greater accuracy. Deep learning models have received considerable attention in stock prediction research due to their superior representational power and lack of undue assumptions. These models are usually based on Convolutional Neural Networks (CNNs) (Chen et al., 2021), Long Short-Term Memory (LSTM) (Banik et al., 2022), Gated Recurrent Units (GRUs) (Gupta et al., 2022), or Attention Mechanisms (Wang et al., 2023). Furthermore, research has indicated that several input variables can account for variations in stock prices. Analyzing the relationships between these variables makes it possible to identify factors that influence stock price fluctuations, thus making multi-input variable models generally more accurate than single-input variable models (Lin et al., 2021).

The proposal for Graph Convolution Networks (GCNs) (Kipf & Welling, 2016) introduces an innovative perspective on constructing inter-variable relationships. The basic idea of GCNs is to explicitly or implicitly build a graph representing the relationships between samples. The graph structure and node features are then utilized as inputs, enabling the capture of complexity through ensemble information and nonlinear transformations, ultimately generating new features. To better depict the interrelationships between variables, we construct a correlation graph wherein each variable serves as a node. The historical data of each variable acts as a node feature, and the relationships between variables are represented as edges. GCN is

* Corresponding author.
E-mail addresses: 2021410057@sdtbu.edu.cn (D. Ma), yuanda@sdtbu.edu.cn (D. Yuan).

https://doi.org/10.1016/j.eswa.2024.123948
Received 23 October 2023; Received in revised form 7 April 2024; Accepted 8 April 2024
Available online 11 April 2024

employed to incorporate these relationships as incremental information into the model, capturing nonlinear relationships between variables and improving predictions of interactions among multivariables.

Given the capacity of GCNs to preserve temporal relationships among variables while capitalizing on their interdependence, the integration of GCNs into the modeling of stock series emerges as a promising avenue for research. A multitude of GCN-based investigations has focused on analyzing the correlation between stocks. Notably, in a study (Chen et al., 2018), an exhaustive graph was constructed, encompassing all involved corporations, based on factual market investment data. By employing node embedding methods on the graph, a distributed representation for each corporation was acquired. An alternative and innovative approach was introduced by Li et al. (2021), leveraging overnight news to establish a relational graph between stocks and employing a GCN to model the connections between stocks. In a related study (Ma et al., 2023), a GCN was implemented to capture the impact of news related to associated companies on the target stock. Building upon these initiatives, Ma et al. (2024) further extended the research by constructing multiple relationship graphs using correlation coefficients and dynamic time warping based on historical stock data.

The relationships among stocks are inherently characterized by volatility, noise, ambiguity, and dynamic changes, often resulting in inaccurate or incomplete stock relationship graphs. In contrast, correlations among transaction data demonstrate greater stability and well-defined patterns. However, the primary emphasis of GCN-based research is directed towards investigating correlations among stocks, and there is a dearth of studies that effectively employ GCNs to harness correlations among transaction data. Furthermore, in stock price prediction, the ability to accurately predict trends is of equal importance. The precision of trend prediction significantly influences the effectiveness of investment decisions and risk management. Focusing solely on numerical precision may compromise the model's ability to capture stock price trends accurately, thereby limiting its utility and reliability. Therefore, it is essential to balance numerical precision with trend accuracy for stock price prediction. However, existing models often emphasize numerical precision while neglecting to accurately capture stock price trends.

To construct spatio-temporal relationships between transaction data and balance the numerical precision and trend accuracy of stock price prediction, this paper introduces a regularized ensemble framework with graph convolution networks. The inherent noise in stock price data, attributed to market conditions and economic fluctuations (Cheng & Wei, 2014), poses a barrier to directly predicting raw time series. Therefore, an ensemble graph convolution network is devised by integrating Variational Mode Decomposition (VMD) and multiple multi-graph convolution networks. VMD is employed to extract features from each dimension of the original transaction data series, representing distinct time scales. These features serve as node attributes in the subsequent graph model. Subsequently, node embeddings are generated individually for features from each dimension of the sequence, utilizing multiple graph convolutional networks to capture relationships within transaction data from various perspectives. Temporal relationships of the node embeddings are modeled using Gated Recurrent Units (GRUs), and the outputs from each sequence dimension are integrated to facilitate stock price prediction.

To strike a balance between numerical precision and trend accuracy in stock price prediction, a trend accuracy regularization term is incorporated into the loss function of the stock price prediction model. This innovative approach enhances the overall performance, practicality, and reliability of the model, making it more suitable for real investment decisions and yielding higher returns. Additionally, the paper introduces an inventive investment strategy that more accurately reflects performance differences among stock price prediction models in trading simulations. This strategy allocates proportional investments to the top k predicted stocks based on both accuracy and return scores.

- A regularized ensemble framework with graph convolution networks, and an investment strategy tailored to the characteristics of stock price prediction are proposed, and the effectiveness of the proposed model and investment strategy are demonstrated on several real-world datasets.
- Introducing an ensemble graph convolutional network framework. The framework employs VMD for extracting features from various time scales within the original time series data. In addition, it utilizes multi-graph convolutional networks to capture concealed correlations across different transaction data scales.
- Incorporating a trend accuracy regularization term into the loss function of the stock price prediction model enhances its overall performance, practicality, and reliability, thereby increasing its suitability for real investment decisions and improving returns. This approach's efficacy is corroborated through trading simulations, which demonstrate its potential to elevate the model's return rate.

The remainder of this paper is organized as follows: Section 2 provides a brief overview of relevant prior research. Section 3 presents a comprehensive description of our model. Section 4 outlines the dataset and details the experimental setup used in our study, accompanied by a presentation and analysis of our experimental results. Finally, Section 5 provides a summary of our work and an outlook for future research.

2. Related work

Stock prediction is a prominent challenge within the fields of time series analysis and machine learning, garnering considerable attention from researchers. These prediction techniques are broadly classified into two categories: statistical and Artificial Intelligence (AI) methods. Statistical methods traditionally assume linearity among normally distributed variables (Wang et al., 2011), employing models such as Linear Regression (LR) (Xu et al., 2022), Autoregressive Conditional Heteroskedasticity (ARCH) models (Alam et al., 2013), Autoregressive Integrated Moving Average (ARIMA) (Kumar et al., 2022), and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) (Liu & Shi, 2022). These approaches are predicated on the notion that the time series in question are generated by a linear process and endeavor to model the underlying process based on this assumption. However, stock data are inherently characterized by high noise levels, nonlinearity, and nonparametric traits (Si & Yin, 2013), rendering statistical methods less effective for stock prediction.

Subsequently, with the boom in artificial intelligence technology, machine learning techniques were introduced to deal with complex financial market data and were shown to be useful for making stock predictions (Tang et al., 2019). In contrast to statistical methods, machine learning models require fewer assumptions and possess robust capabilities for nonlinear mapping, enabling them to accommodate a plethora of features. Common machine learning methodologies encompass Random Forest (RF) (Park et al., 2022), Decision Tree (Qiu et al., 2022), Support Vector Regression (SVR) (Ye et al., 2022), and neural networks (Lv et al., 2022). Notably, the advent of deep learning has gained immense popularity in stock prediction, enhancing the capacity to capture complex data representations. Widely adopted deep learning methods include Convolutional Neural Networks (CNNs) (Chen et al., 2021), Recurrent Neural Networks (RNNs) (Dudukcu et al., 2023), Long Short-Term Memory networks (LSTMs) (Banik et al., 2022), Restricted Boltzmann Machines (RBMs) (Xi & Xu, 2023), and Gated Recurrent Units (GRUs) (Gupta et al., 2022).

To enhance the efficacy of stock prediction models in capturing the intricate and nonlinear relationships prevalent in the stock market, researchers have increasingly turned to multivariate prediction methods. These methods incorporate diverse data types such as opening prices, high prices, low prices, and trading volumes into multivariate models, thereby augmenting the accuracy and generalizability of stock

predictions (Lin et al., 2021). For instance, in Park et al. (2022), a multi-task model was constructed by integrating multiple technical indicators to enhance predictability and profitability. Similarly, Chen et al. (2022) employed Dynamic Time Warping (DTW) for distance calculation and K-means clustering to group stock price data, leveraging the interactions within clusters to improve prediction accuracy.

Furthermore, Graph Convolutional Networks (GCNs) have been introduced to enhance the capture of correlations among variables, particularly in applications such as stock prediction. Chen et al. (2021) proposed a novel method for stock trend prediction using a Graph Convolutional feature-based Convolutional Neural Network (GC-CNN) model, which considers both stock market information and individual stock details. In a related study (Ma et al., 2023), a GCN was employed to capture the news effects of associated companies on the target stock. The methodology involved integrating numerical characteristics, market-driven news sentiment, and the news sentiment of related stocks to predict stock price movements. Ma et al. (2024) extended the research by constructing multiple relationship graphs using correlation coefficients and dynamic time warping based on historical stock data. Their predictive model incorporated decomposition algorithms, graph convolutional networks, and adversarial training techniques for forecasting stock prices. Additionally, Feng et al. (2022) applied normalized mutual information based on a logarithmic rate of return and Detrended Cross-Correlation Analysis (DCCA) coefficients to assess stock market correlations, affirming the utility of graphical models in finance. Cui, Li et al. (2023) proposed the HyperGraph Tri-Attention Network (HGTAN) to enhance hypergraph convolutional networks with a hierarchical organization of intra-hyperedge, inter-hyperedge, and inter-hypergraph attention modules. HGTAN adaptively determines the importance of nodes, hyperedges, and hypergraphs during information propagation among stocks, enabling the full exploitation of potential synergies between stock movements. Moreover, Li et al. (2022) proposed a novel similarity framework to overcome the limitations of chart similarity on specific charts. Specifically, after extracting the key point sequence from the stock price sequence, it is transformed into a graph, and ultimately, any graph kernel, such as the Weisfeiler Lehman kernel and GCN, is used to fully mine the information in the graph to predict stock trends.

To mitigate the impact of stock data's nonlinear, non-stationary properties, and inherent noise on model performance, hybrid models combining decomposition algorithms with predictive models have emerged, showing promising superiority. These hybrid models employ decomposition algorithms to break the original time series into relatively smooth subsequences, facilitating more efficient construction of predictive models. In one instance, Ma et al. (2024) decomposed subseries using Variational Modal Decomposition (VMD) and used them as inputs to a multi-graph convolutional network, optimizing VMD parameters with Genetic Algorithms (GA) to predict stock prices for the subsequent day. Another approach employs Multivariate Empirical Modal Decomposition (MEMD) coupled with Long Short-Term Memory (LSTM) models for multi-step-ahead prediction of stock price indices, capitalizing on the intrinsic characteristics of complex stock price index time series (Deng et al., 2022). Furthermore, Cui, Wang et al. (2023) introduced a multichannel VMD-CBAM-BiLSTM hybrid model, termed McVCsB, which combines VMD and Convolutional Block Attention Module (CBAM) algorithms for noise reduction and deep feature extraction, employing an improved decomposition-reconstruction prediction framework for stock index prediction.

In summary, based on current knowledge, the primary emphasis of GCN-based research is directed towards investigating correlations among stocks, and there is a dearth of studies that effectively employ GCNs to harness correlations among transaction data. Moreover, the ability to accurately predict trends is equally important in stock price forecasting. The exclusive emphasis on numerical precision poses a risk of compromising the model's capacity to capture stock price trends, thereby constraining its utility and reliability. Despite this, prevalent stock price prediction models persistently prioritize numerical precision, often neglecting the nuanced capture of stock price trends.

3. Methodology

3.1. Problem formulation

The stock price prediction task can be defined as a multivariate time series prediction problem. In this study, we employ transaction data related to a stock, including attributes such as open price, close price, high price, low price, and trading volume, to predict the close price of the stock. To better leverage the relationships among transaction data, our research constructs multiple weighted prior knowledge graphs through measures of similarity and distance among transaction data, denoted as $\mathcal{G} = (G_1, G_2, \ldots, G_m)$. Here, $G_i = (V_i, E_i)$, where V_i signifies a set of $N = |V_i|$ vertices representing distinct transaction data points, and E_i represents a set of edges characterizing the strength of relationships between these vertices.

In this work, we denote the state of a stock at time step T as a feature vector $S_T = (s_0, s_1, \ldots, s_{T-1})$, where s_i corresponds to various time intervals (typically daily in this context). The stock's state at time t is represented as a collection of transaction data features $s_t = (C_{open}, C_{close}, \ldots, C_{volume})$. Therefore, the formulated stock price prediction problem can be defined as follows:

Given a graph G and a sequence of stock transaction data with a window length of L, denoted as $S_L = (s_0, s_1, \ldots, s_{L-1})$, our objective is to learn a function f that utilizes S_L and G to predict the closing price at the next time step, denoted as $\hat{C}^L_{close}$. This can be formally expressed as follows:

$$\hat{C}^L_{close} = f(S_L, \mathcal{G}) \tag{1}$$

3.2. Framework overview

To harness the correlations within transaction data effectively while striking a balance between numerical precision and trend accuracy, this paper introduces a regularized ensemble framework with graph convolutional networks for stock price prediction. The framework of the proposed method is depicted in Fig. 1. The model is structured into five key parts: sequence decomposition, graph generation, multi-graph convolutional network, gated recurrent unit, and regularization loss.

In the sequence decomposition part, the transaction data are decomposed by Variational Modal Decomposition (VMD) to obtain smoother subsequences of different scales. After that, the relationship graphs are constructed by calculating both correlation (Pearson correlation, Spearman's Rank correlation) and distance (dynamic time warping) among the transaction data of each subsequence, respectively. The Multi-Graph Convolutional Network (Multi-GCN) part treats each transaction data point as a node within the graph and generates node embeddings to capture spatial relationships between nodes. Next, the Gated Recurrent Unit (GRU) is employed to model the temporal dependencies of the node embeddings. The predicted values for each subsequence are aggregated to produce the predicted value for the original sequence. To strike a balance between numerical precision and trend accuracy, this paper introduces regularization to the Mean Squared Error (MSE) loss and incorporates a regularization term to improve the overall performance of the prediction.

3.3. Sequences decomposition

Stock price series exhibit considerable uncertainty and noise (Cui, Wang et al., 2023), which renders stock data more challenging to predict. Employing decomposition algorithms can decompose the original sequences into multiple smoother subsequences, thereby reducing the effects of data nonstationarity and noise. Furthermore, the integration of decomposition algorithms into stock prediction models has demonstrated effectiveness in enhancing model performance (Lahmiri, 2016). Common decomposition techniques encompass wavelet decomposition,

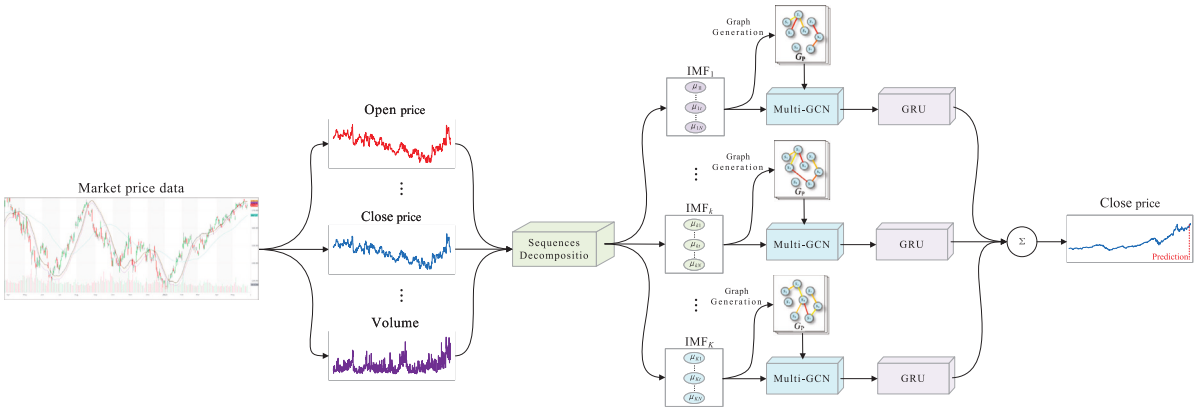


Fig. 1. Framework of the proposed method.

Table 1
Results of the ADF test for the original series and the decomposed IMFs.

	Original	IMF1	IMF2	IMF3
ADF statistic	−0.8875	−1.6647	−7.3048	−13.2007
p-value	0.792055	0.449410	0.0000	0.0000

Empirical Modal Decomposition (EMD), Singular Spectrum Analysis (SSA), and Variational Modal Decomposition (VMD). Among these, EMD and VMD are the most widely adopted and effective methods in the field of stock prediction (Deng et al., 2022; Huang et al., 2021).

VMD is an adaptive decomposition method proposed by Dragomiretskiy and Zosso (2013), which uses a non-recursive signal decomposition optimization algorithm. Compared to EMD, VMD exhibits greater efficiency and lower noise sensitivity in data processing. Consequently, VMD proves more suitable for handling complex data and has shown superior performance in financial data forecasting (Lahmiri, 2016).

By obtaining the optimal solution of the constrained variational model, VMD achieves the decomposition of the original sequence $f(t)$ into K Intrinsic Modal Functions (IMFs), μ_k, with each μ_k intended to closely match a center frequency ω_k. The bandwidths of these modes can be estimated through the following steps: first, for each mode μ_k, the Hilbert transform is used to compute the correlation analysis signal, and then the one-sided spectrum is derived. Second, the spectrum of each mode μ_k is shifted to the baseband using exponential mixing, with the center pulsation tuned to the respective estimate. Finally, the bandwidth is estimated via H1 Gaussian smoothing of the demodulated signal. The constraints of the variational problem can be expressed as follows:

$$\min_{\{\omega_k\},\{\mu_k\}} g(\omega_k,\mu_k) = \min_{\{\omega_k\},\{\mu_k\}} \sum_{k=1}^{K} \left\| \partial_t[(\delta(t)+j/(\pi t))*\mu_k(t)]\exp(-j\omega_k t)\right\|_2^2 \tag{2}$$

where $\{\mu_k\} = \{\mu_1, \mu_2, \ldots, \mu_k\}$ and $\{\omega_k\} = \{\omega_1, \omega_2, \ldots, \omega_k\}$ denote the set of kth submodules and their corresponding center frequencies, respectively. ∂_t represents the time derivative, $\delta(t)$ is the Dirac function, $*$ denotes convolution, and K is the number of IMFs.

To optimize the constrained results, quadratic penalty parameters and augmented Lagrange equations are introduced. The constrained optimization problem can be expressed as follows:

$$L(\{\omega_k\},\{\mu_k\},\lambda) = \alpha g(\omega_k,\mu_k) + \left\| f(t) - \sum_{k=1}^{K}\mu_k(t)\right\|_2^2 + \langle \lambda(t), f(t) - \sum_{k=1}^{K}\mu_k(t)\rangle \tag{3}$$

where α denotes the balancing parameter for the data fidelity constraint.

Table 1 presents the results of the Augmented Dickey–Fuller (ADF) test conducted on the original stock sequence and the decomposed Intrinsic Mode Functions (IMFs) in the SSE dataset with the closing price of 600,000 stock codes. Smaller values of the ADF statistic and *p*-value in Table 1 indicate improved series stability. The table's data highlights the effective reduction of non-stationarity through VMD decomposition.

The parameter settings for VMD include the number of IMFs (K), the penalty factor (α), the Lagrange multiplier (τ), and the convergence tolerance level (ϵ). Notably, τ, and ϵ have a limited impact on the decomposition results and are often set as constants. In contrast, K and α significantly affect the decomposition outcomes and are typically determined based on the characteristics of the data.

Genetic Algorithm (GA) (Holland, 1992) is a versatile meta-heuristic algorithm known for its adaptability, global search capability, and scalability. It operates by exploring the solution space of a problem and progressively optimizing solutions through operations such as selection, crossover, and mutation, simulating genetic processes. GA finds extensive application in optimization problems, search problems, and machine learning due to its flexibility and suitability for handling complex problem spaces and multi-objective optimization challenges.

Based on the findings in the literature (Jia et al., 2021), the Pearson coefficient between residuals and the original sequence should be minimized when employing VMD for complete decomposition. Based on this theory, this study employs a combined optimization approach that integrates GA and residual similarity to determine the optimal K and α. In addition, the training set is initially decomposed, and the same parameters are then applied to decompose the test set to prevent data leakage and maintain consistency. Subsequently, the fitness function for GA is defined as the average Spearman's Rank correlation coefficient between the original sequence and the residuals of the training set, denoted as $\bar{\rho}_{Sp}(f_{\text{train}}(t), \text{Res}_{train}(t))$. The combined algorithm can be expressed as follows:

$$min\bar{\rho}_{Sp}(f_{\text{train}}(t), \text{Res}_{train}(t)) = \frac{\sum_{n=1}^{N}\rho_{Sp}(f^n_{\text{train}}(t), \text{Res}^n_{train}(t))}{N}$$

$$s.t. \begin{cases} K \in \{2,3,\ldots,10\} \\ \alpha \in \left[\frac{Size\left(f^n_{train}(t)\right)}{2},\ 3\times Size\left(f^n_{train}(t)\right)\right] \end{cases}$$

where N represents the number of types of transaction data, $f_{\text{train}}(t)$ signifies the original training set sequence of the transaction data, and Size($f_{\text{train}}(t)$) represents the length of the original sequence in the training set.

3.4. Graphs generation

To better leverage correlations among variables, researchers have introduced Graph Convolutional Networks (GCNs) to model spatial relationships. Most existing stock prediction methods based on GCNs focus on capturing correlations among stocks, while lack of research on

the correlation of transaction data. In reality, in contrast to the volatile, ambiguous, and dynamically changing correlations among stocks, correlations among transaction data are more stable and explicit, and more suitable for stock prediction.

Therefore, in this study, stock transaction data are viewed through a graph-based perspective, where nodes represent individual transaction data, and edges signify hidden dependencies between these data. Consistent with the approach in prior literature (Ma et al., 2024), this paper employs both the Pearson correlation coefficient and Spearman's Rank correlation coefficient between transaction data to construct synchronous correlation graphs. In addition, it calculates Dynamic Time Warping (DTW) between transaction data to construct asynchronous correlation graphs. By combining multiple graphs, the correlations among the transactional data can be represented more comprehensively. Specifically, for each type of relation $r \in R$, a weighted undirected graph $G_r = (V, E)$ is constructed, with each node $v \in V$ representing one type of transaction data. The graph G_r is built from the adjacency matrix $A_r = (a_{ij})_{N\times N}$, where a_{ij} denotes the strength of the correlation between the ith and jth transaction data.

Pearson correlation coefficient is a statistical measure that quantifies the linear relationship between two series, while Spearman's Rank correlation coefficient assesses the strength of the monotonic relationship between stock price series. To calculate Spearman's Rank correlation coefficient, the series are sorted in ascending order and assigned ranks. For a sample size of n, Pearson's correlation coefficient and Spearman's Rank correlation coefficient are calculated as follows:

$$\rho_P(U,V) = \frac{E\left[\left(U-\mu_U\right)\left(V-\mu_V\right)\right]}{\sqrt{\sum_{i=1}^{n}\left(U_i-\mu_U\right)^2\sum_{i=1}^{n}\left(V_i-\mu_V\right)^2}} \tag{4}$$

$$\rho_{Sp}(U,V) = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{5}$$

where μ_U and μ_V are the mean values of sequences U and V, respectively; n is the length of the sequence, and d_i is the rank difference between U_i and V_i.

DTW is an algorithm that measures the similarity between two time series by aligning them to find the best alignment. Asynchronous correlation graphs between time series can be constructed using DTW on transaction data. However, the classical DTW algorithm has high time complexity and is unsuitable for large-scale data. Therefore, in this study, we employ the FastDTW algorithm (Salvador & Chan, 2007), an approximation of DTW, to replace the traditional DTW algorithm. Additionally, we normalize the results to facilitate their combination with graphs G_P and G_{Sp}. Denoting the FastDTW algorithm as $f_{DTW}()$, the correlation coefficient is calculated as follows:

$$\rho_{DTW}(U,V) = 1 - \frac{f_{DTW}(U,V)}{n} \tag{6}$$

3.5. Multi-graph convolutional network

Graph Convolutional Networks (GCNs), as one of the common models for capturing relationships, have been used in many studies in stock prediction. To capture the correlation between transaction data more completely and comprehensively, this paper introduces a Multi-Graph Convolutional Network (Multi-GCN). The schematic diagram of Multi-GCN is illustrated in Fig. 2. From the diagram, it becomes evident that synchronous correlations exist among the open price, close price, the high price, and the low price. However, their changes are occasionally asynchronous with variations in trading volume. The Multi-GCN integrates multiple generated graphs into a unified graph, subsequently employing a graph convolutional network to investigate and capture latent relationships among transaction data from diverse perspectives. This approach facilitates a comprehensive analysis of the complex relationships among transaction data. Given that the graphs G_P, G_{Sp} and G_{DTW}, generated by the graph generation component, tend to be sparse, these graphs are densified and fused into a composite graph G_S. The adjacency matrix A_S of the synthesis graph G_S is computed as follows:

$$A_S = \sum_{k=1}^{K} w_k A_k \tag{7}$$

GCN is a deep learning model specifically designed to model and learn from non-regular graph data, and the core idea is to update the representation of a node by aggregating its neighbor information. This aggregation operation resembles the filtering operation in traditional convolutional neural networks but relies on the graph's topology rather than a grid structure. Specifically, GCN calculates a weighted average of each node's features with those of its neighbors to obtain an updated node representation. This weighted averaging process can be realized through matrix operations. To simplify graph convolution, GCN commonly employs a local first-order approximation, computed at each layer as follows:

$$H^{l+1} = ReLU\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^l W^l\right) \tag{8}$$

where $\widetilde{A} = A + I_n$, the diagonal array $\widetilde{D}$ represents the degree matrix with $\widetilde{D}_{ii} = \sum_j \widetilde{A}^{(ij)}$, H^l denotes the node vector of the lth layer, H^0 is the input matrix, and W^l is the training weight for the respective layer.

To simplify notation, define $\hat{A} = \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}$. With this definition, each layer of graph convolution can be expressed as:

$$H^{l+1} = ReLU\left(\hat{A}H^l W^l\right) \tag{9}$$

Multi-GCN facilitates the generation of node embeddings that incorporate node attributes and topology, enabling the subsequent modeling of temporal dependencies in the data.

3.6. Gated recurrent unit

Temporal dependencies of stock are crucial to achieving precise stock price predictions. Consequently, this study incorporates Gated Recurrent Units (GRUs) to model the temporal dependencies of node embeddings. GRUs are a variant of Recurrent Neural Networks (RNNs) used to process sequential data and model temporal dependencies. In comparison to conventional RNNs, GRUs introduce a gating mechanism designed to improve the modeling of long-term dependencies and mitigate the issue of vanishing gradients. The design of GRU is based on the Long Short-Term Memory (LSTM) model, which simplifies the structure of LSTM and reduces the number of parameters while retaining similar modeling capabilities. A pivotal feature of GRUs is the inclusion of two gating units: the reset gate and the update gate.

The reset gate governs how the current moment's inputs are blended with the hidden state from the previous time step. By incorporating the reset gate, GRUs possess the capability to selectively disregard the previous hidden state, thereby enhancing their capacity to handle long-term dependencies. The update gate regulates the retention of the previous time step's hidden state. It dictates the trade-off between preserving the prior moment's hidden state and incorporating the current moment's candidate hidden state. The update gate dynamically learns to selectively update and convey information, thereby aiding the model in capturing essential sequence features.

The computational process of a GRU unit is articulated as follows:

$$r_t = \sigma\left(W_r \cdot x_t + U_r \cdot h_{t-1} + b_r\right) \tag{10}$$

$$z_t = \sigma\left(W_z \cdot x_t + U_z \cdot h_{t-1} + b_z\right) \tag{11}$$

$$\widetilde{h}_t = tanh\left(W_h \cdot x_t + r_t \circ (U_h \cdot h_{t-1}) + b_h\right) \tag{12}$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \widetilde{h}_t \tag{13}$$

where σ represents the sigmoid function, r_t and z_t correspond to the reset gate and update gate, respectively. h_{t-1} and x_t denote the hidden

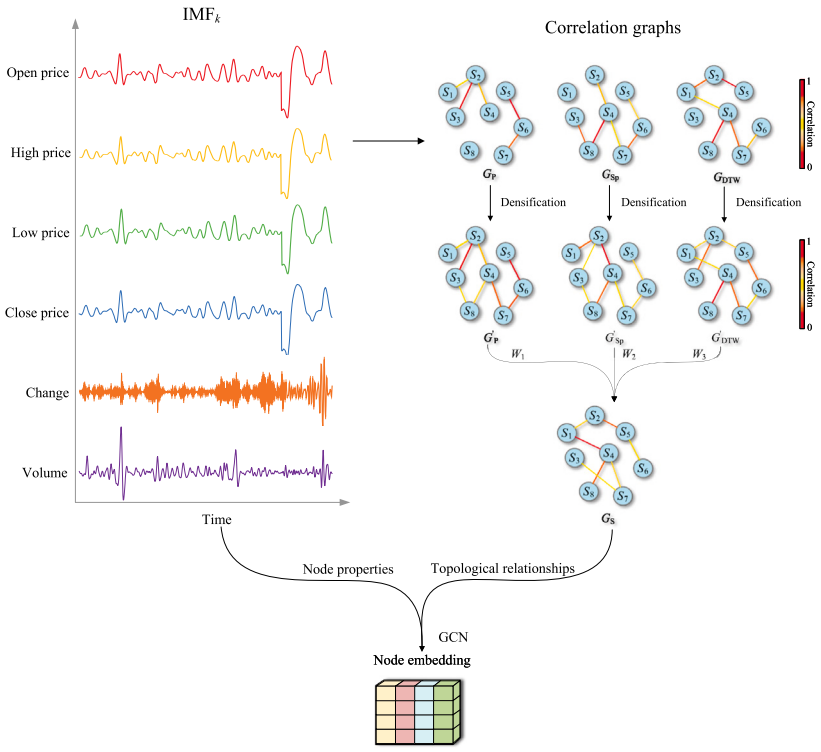


Fig. 2. Schematic diagram of multi-graph convolutional network.

Table 2
Features of the datasets.

Begin date	End date	Dataset	Train	Valid	Test	Total	Feature number	Stock number
4/1/2021	29/12/2023	SSE	508	72	145	726	6	40
		DJIA	526	75	150	752	9	28

layer state at time $t-1$ and the node embedding of the inputs at time t, respectively. $\widetilde{h}_t$ represents the candidate for storing the input values, and $\circ$ denotes the Hadamard product. The symbols W, U, and b represent weight matrices and a bias vector, respectively.

3.7. Regularization loss

In stock price prediction, both numerical precision and trend accuracy hold significant importance. Numerical precision gauges the disparity between predicted and actual values, while trend consistency assesses the ability to forecast price movement directions. The precision of stock price forecasts plays a pivotal role in shaping investment outcomes. A substantial deviation between predicted and actual values can lead to erroneous investment decisions and financial losses. Furthermore, grasping market trends is crucial. Stock market price movements tend to exhibit certain trends, such as upward or downward movements. Effectively capturing these trends aids investors in devising strategic trading approaches and enhancing their investment returns. Focusing solely on minimizing numerical precision may result in a model that predominantly predicts absolute price values, potentially overlooking the directional trends in price movements. Conversely, an excessive emphasis on trend consistency might disregard specific price values, potentially causing missed investment opportunities. Consequently, it is imperative to strike a balance between numerical precision and trend consistency.

To achieve an equilibrium between numerical precision and trend accuracy, we introduce L1 regularization measures to the Mean Squared Error (MSE) loss and incorporate a regularization term for trend consistency. Regularizing the MSE loss diminishes the model's sensitivity to noise and outliers, concurrently enhancing control over the average forecast errors. The introduction of a regularization term encourages the model to acquire smoother prediction functions, effectively mitigating the risk of overfitting. Additionally, introducing an L1 regularization term for trend accuracy can better capture stock price trend changes. This regularization term incentivizes the model to predict accurate price trends, thereby elevating the overall quality of stock price prediction by penalizing incorrect trend predictions.

The ultimate loss function encompasses both the regularization term for the MSE loss and the regularization term for trend accuracy. We employ a trade-off coefficient to calibrate the relative significance of these two regularization components, tailoring them to the specific requirements of various application scenarios. This approach enables us to strike a harmonious balance between the precision of numerical predictions and the consistency of price trend forecasts in stock price prediction, thereby enhancing the overall performance of the model. The ultimate loss function can be expressed as follows:

$$mse_{loss} = \frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2 \tag{14}$$

$$Loss = l1(r_{MSE}) \cdot (mse_{loss}) + l1(r_{ACC}) \cdot (sign(Y_n - X_{n-1}) - sign(X_n - X_{n-1})) \tag{15}$$

where X_i represents the true stock price, Y_i denotes the predicted stock price, n signifies the total number of samples, r_{ACC} and r_{MSE} are regularization coefficient.

4. Experiments

This section encompasses the experimental setup, results, and subsequent analysis. Specifically, Section 4.1 furnishes detailed information regarding the dataset employed, while Section 4.2 expounds upon the comprehensive experimental setup. Section 4.3 effectively showcases the predictive performance of the proposed method in contrast to the

Table 3
Stock codes included in the datasets.

SSE	600000	600015	600016	600018	600028	600036	600048
	600050	600104	600111	600585	600637	600795	600837
	600887	601006	601088	601166	601169	601186	601211
	601288	601318	601328	601336	601398	601601	601628
	601668	601669	601688	601766	601800	601818	601857
	601901	601985	601988	601989	601998		
DJIA	AAPL	AMGN	BA	BA	CAT	CRM	CSCO
	CVX	DIS	GS	HD	HON	IBM	INTC
	JNJ	JPM	KO	MCD	MMM	MRK	MSFT
	NKE	PG	TRV	UNH	V	VZ	WBA
	WMT						

baseline model. An in-depth scrutiny of the model components' influence on prediction performance is presented through an ablation study in Section 4.4. Furthermore, Section 4.5 delves into the examination of the model's sensitivity to hyperparameters. Lastly, Section 4.6 conducts a trading simulation to evaluate the effectiveness of the proposed approach.

4.1. Dataset description

To assess the efficacy of the proposed methodology, experimental evaluations were conducted utilizing two authentic stock market datasets obtained from Ref. Liu, Ma et al. (2022). The particulars of these datasets are delineated below.

(a) SSE Dataset: This dataset encompasses daily transaction data spanning from January 4, 2021, to December 29, 2023, encompassing stocks within the SSE 50 index in the Chinese market. It includes nine categories of transaction data: opening price, closing price, high price, low price, rounded closing price, gain/loss, true gain/loss, trading volume, and turnover. Access to this dataset is facilitated through Tushare.

(b) DJIA Dataset: This dataset encapsulates daily transaction data for stocks listed in the Dow Jones Industrial Average within the U.S. market, spanning from January 4, 2021, to December 29, 2023. It encompasses six types of transaction data: opening price, closing price, high price, low price, rounded closing price, and trading volume. The dataset is obtainable from Yahoo Finance.

In order to uphold the integrity and precision of our results, stocks featuring incomplete price data and outliers were systematically excluded from the analytical framework. Subsequently, the datasets underwent meticulous partitioning into training, validation, and test sets, adhering to a prescribed ratio of 7:1:2. A detailed breakdown of this dataset division is outlined in Table 2.

The datasets consist of a range of stock symbols covering the period from January 4, 2021, to December 29, 2023. A comprehensive list of these stock symbols is presented in Table 3. Throughout the specified timeframe, the SSE dataset encompasses 40 ticker symbols, while the DJIA dataset includes 28 ticker symbols.

4.2. Experimental setup

4.2.1. Data preprocessing

To minimize the influence of noise, the dataset underwent Min-Max normalization utilizing the following normalization formula:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{16}$$

To prevent any compromise to model credibility arising from the data leakage issue, a distinct sequence decomposition and normalization process was carried out for both the training set and the test set. Furthermore, inverse normalization was applied before computing the evaluation metrics to ensure that the data provides a realistic assessment of model performance and quantifies the difference between predicted and actual values.

4.2.2. Evaluation metrics

In this paper, the prediction accuracy of the model is assessed using commonly employed metrics, including Accuracy (ACC), R-squared (R^2), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). These metrics are calculated as follows:

$$ACC = \frac{n_{correct}}{n} \tag{17}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(X_i - Y_i)^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2} \tag{18}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2} \tag{19}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|X_i - Y_i| \tag{20}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{X_i - Y_i}{X_i}\right| \times 100 \tag{21}$$

where X_i represents the true stock price, Y_i denotes the predicted stock price, $\bar{X}$ is the average of the true stock prices, n signifies the total number of samples, and $n_{correct}$ stands for the number of samples with correctly predicted trends.

In general, higher values of ACC and R^2, as well as lower values of RMSE, MAE, and MAPE, indicate superior predictive performance of the model.

4.2.3. Baseline methods

To assess the forecasting performance of the proposed method, a comparative analysis was conducted with several established methods, including ARIMA, GRU, PF-LSTM, GCGRU, CEEMDAN-BiLSTM, SCINet, StemGNN, Dlinear, Fedformer, N-hits, WaveForM, FourierGNN and VGC-GAN. These methods are described as follows:

- **ARIMA** (Salvador & Chan, 2007): ARIMA, a classical time series forecasting method, combines the concepts of autoregression (AR) and moving average (MA) to analyze and forecast trends and seasonal variations in time series data.
- **GRU** (Chung et al., 2014): GRU, a variant of RNNs with a gating mechanism, effectively resolves the vanishing or exploding gradient issue in handling long-term dependencies. GRU is commonly employed for stock forecasting.
- **PF-LSTM** (Ma et al., 2020): PF-LSTM combines RNNs and particle filtering by learning potential particle belief representations and updating them using particle filters. In this paper, PF-LSTM serves as a direct alternative to LSTM.
- **GCGRU** (Yin et al., 2021): GCGRU integrates GCNs and GRUs. It constructs a stock relationship graph by calculating Pearson correlation coefficients from historical stock data and extracts features from the price of each stock and the prices of those highly similar stocks in the graph using GCNs. Subsequently, a sequence of these extracted features is fed into a GRU model to capture temporal dependence.
- **CEEMDAN-BiLSTM** (Lin et al., 2022): The CEEMDAN-LSTM model integrates LSTM and complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) for forecasting the Realized Volatility (RV) of the CSI300, S&P500, and STOXX50 indices. In this study, BiLSTM serves as an upper-level substitute for LSTM, contributing to enhanced model performance.
- **SCINet** (Liu, Zeng et al., 2022): SCINet utilizes a hierarchical downsampling-convolution-interaction structure equipped with a diverse set of convolutional filters. It iteratively extracts and exchanges information at various temporal resolutions, enhancing predictability through efficient representations.

- **StemGNN** (Liu, Zeng et al., 2022): StemGNN is a multivariate time series forecasting model that amalgamates Graph Fourier Transform (GFT) and Discrete Fourier Transform (DFT) to capture inter-series correlations and temporal dependencies. Leveraging convolution and sequential learning modules, StemGNN effectively predicts spectral representations, autonomously learning inter-series correlations without predefined priors.
- **DLinear** (Zeng et al., 2023): DLinear decomposes time series into trend and cosine components and subsequently employs two single-layer linear networks to model these components for forecasting.
- **Fedformer** (Zhou et al., 2022): Fedformer merges the Transformer with seasonal trend decomposition methods for time series analysis, offering linear time complexity, making it more efficient than Transformer.
- **N-hits** (Challu et al., 2022): N-hits extends the N-BEATS model to enhance forecasting accuracy and computational efficiency. Techniques such as hierarchical interpolation and multi-rate data sampling contribute to these improvements.
- **WaveForM** (Yang et al., 2023): WaveForM presents an end-to-end framework for multivariate time series prediction. It combines discrete wavelet transform with graph construction and utilizes a graph-enhanced prediction module featuring expansion convolution and graph convolution to capture correlations and predict wavelet coefficients at different levels.
- **FourierGNN** (Yi et al., 2023): Addressing the challenge of multivariate time series prediction, FourierGNN treats each sequence value (irrespective of variable or timestamp) as a graph node. It directly applies a graph network to capture inter-sequence dynamics and intra-sequence correlations. The model introduces the concept of hypervariable graphs and utilizes the Fourier Graph Operator (FGO) to perform matrix multiplication in Fourier space.
- **VGC-GAN** (Ma et al., 2024): The VGC-GAN model employs a multi-graph convolutional adversarial framework to predict stock prices by generating multiple correlation graphs from historical stock data, comprehensively capturing inter-stock correlations. It combines a multi-graph convolutional network and GRU within a Generative Adversarial Network (GAN) framework, trained with a CNN to explore hidden correlations between stocks and time dependence. Additionally, the model incorporates VMD to mitigate the impact of noise.

4.2.4. Parameter settings

In this experimental study, a series of trials were conducted to determine the most effective hyperparameters. Following evaluation, the optimal configuration was identified, and the selected hyperparameters for the subsequent experiments are as follows: a learning rate of 0.008, 100 epochs, a batch size of 128, a sliding window length (L) set to 30, and regularization coefficients (r_{MSE} and r_{ACC}) assigned values of 0.01 and 0.1, respectively. The optimal values for the sliding window length (L) and the regularization coefficients (r_{MSE} and r_{ACC}) are comprehensively discussed in Section 4.5, providing insights into the careful selection process that led to their determination. For the application of VMD in this study, the optimal parameters (K and α) specific to the SSE and DJIA datasets are presented in Table 4.

Table 4
Optimal VMD parameters in datasets.

Dataset	Stock code	K	α	Stock code	K	α	Stock code	K	α
SSE	600000	3	296	600887	3	315	601628	3	310
	600015	3	331	600958	4	387	601668	3	347
	600016	3	331	601006	4	298	601669	3	313
	600018	4	371	601088	4	311	601688	4	391
	600028	4	336	601166	4	293	601766	3	380
	600036	4	404	601169	4	308	601800	3	334
	600048	4	373	601186	3	344	601818	3	355
	600050	3	305	601211	3	299	601857	3	335
	600104	4	339	601288	4	302	601901	4	307
	600111	4	402	601318	3	330	601985	4	372
	600585	3	367	601328	3	307	601988	4	292
	600637	3	314	601336	3	292	601989	4	300
	600795	4	575	601398	3	306	601998	4	360
	600837	4	376	601601	3	415			
DJIA	AAPL	3	417	HD	4	334	MSFT	3	314
	AMGN	4	415	HON	3	422	NKE	3	357
	AXP	3	336	IBM	4	304	PG	4	366
	BA	4	389	INTC	4	346	TRV	4	317
	CAT	4	342	JNJ	4	324	UNH	4	305
	CRM	3	339	JPM	2	337	V	4	386
	CSCO	3	327	KO	2	331	VZ	4	353
	CVX	4	602	MCD	4	306	WBA	3	334
	DIS	3	345	MMM	4	372	WMT	4	324
	GS	4	348	MRK	4	310			

4.3. Performance analysis

To assess the effectiveness of the proposed method, a comparative analysis was conducted with a baseline approach utilizing the same dataset. The corresponding results are presented in Table 5, while Fig. 3 provides a visual representation of the comparative predictive performance between the proposed method and the baseline methods.

Upon careful examination of the data in Table 5, along with insights derived from Fig. 3, it is evident that the proposed model consistently demonstrates robust performance across all time periods in both datasets. The proposed method achieves superior values in all assessed metrics, with significant improvements in ACC and R^2 by more than 15% and 10%, respectively. Furthermore, the RMSE, MAE, and MAPE exhibit reductions exceeding 40% when compared to the sub-optimal method. These outcomes underscore the effectiveness, robustness, and strong adaptability of the proposed method.

4.4. Ablation studies

To explore the impact of different modules on model prediction performance, the proposed method underwent modifications, resulting in nine variants. Table 6 outline the performance of these variants alongside the proposed model, maintaining consistent experimental configurations across all approaches.

Among these variants, Methods 1, 2, 3, and 4 employed EMD, EEMD, CEEMDAN, and SSA, respectively, instead of VMD for sequence decomposition. These methods share identical components and parameters with the proposed approach, with the exception of their distinct decomposition algorithms. Notably, all methods incorporate the use of maximum–minimum normalization. The decomposition algorithms employed in the comparative methods are elucidated below:

1. Empirical Mode Decomposition (EMD):

EMD is a data-driven technique designed to decompose a given time series into intrinsic mode functions (IMFs) based on local data characteristics. Each IMF signifies a narrowband oscillatory component with varying scales. EMD has garnered widespread adoption across diverse domains, including signal processing and time series analysis.

2. Ensemble Empirical Mode Decomposition (EEMD):

An extension of EMD, EEMD addresses issues such as mode mixing and end effects by introducing noise to the original data. Multiple EMD decompositions are performed on the noisy realizations, and the resulting IMFs are averaged to mitigate mode mixing, thus enhancing the overall decomposition results.

3. Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN):

Building upon the EEMD framework, CEEMDAN further refines the approach by adaptively determining the noise level for each ensemble realization. This adaptive noise inclusion contributes to a more precise decomposition by diminishing mode mixing and preserving the intrinsic properties of the data.

4. Singular Spectrum Analysis (SSA):

Table 5
Predictive performance of the proposed method and baseline methods. Results with **bold** are the best performance.

Method	SSE					DJIA				
	ACC	R^2	RMSE	MAE	MAPE	ACC	R^2	RMSE	MAE	MAPE
ARIMA	0.5036	−5.1041	1.3242	1.0976	9.972%	0.5007	−2.5318	16.3622	13.4835	8.475%
GRU	0.5097	−2.0921	0.7328	0.5913	5.076%	0.4664	−2.2375	15.8586	14.1495	5.665%
PF-LSTM	0.5906	0.5777	0.3704	0.2964	2.023%	0.5783	0.5692	4.5564	3.8222	2.905%
GCGRU	0.4937	0.5594	0.3149	0.2557	1.774%	0.4628	−42.4816	69.8978	68.8906	26.320%
CEEMDAN-BiLSTM	0.5916	0.6838	0.2825	0.2289	1.654%	0.5581	0.3065	6.2354	4.9513	2.831%
SCINet	0.5171	−1.7009	0.7886	0.6326	4.941%	0.4775	−29.1912	43.7255	42.5843	23.947%
StemGNN	0.5179	−2.9559	0.8587	0.7589	5.785%	0.4932	−0.4291	5.9429	4.8690	4.464%
DLinear	0.4990	0.8365	0.2547	0.1924	1.420%	0.5307	0.8727	2.7149	2.3394	1.521%
FEDformer	0.4841	−0.2689	0.5150	0.4044	3.462%	0.4803	0.3743	5.5055	4.4107	3.049%
N-hits	0.5164	−8.6022	2.8734	2.6278	28.753%	0.4927	−5.0808	20.5275	17.5930	10.557%
WaveForm	0.4772	−23.7399	1.7128	1.5182	12.539%	0.4818	−3.3862	55.3529	53.5810	57.897%
FourierGNN	0.4884	−1.2027	0.9438	0.8208	5.667%	0.4683	−1.4609	14.6036	12.2370	7.221%
VGC-GAN	0.5936	0.5601	0.2569	0.1894	1.409%	0.5815	0.7715	2.6791	2.0497	1.216%
Proposed	**0.6940**	**0.9226**	**0.1500**	**0.1137**	**0.843%**	**0.7311**	**0.9652**	**1.6082**	**1.2206**	**0.714%**

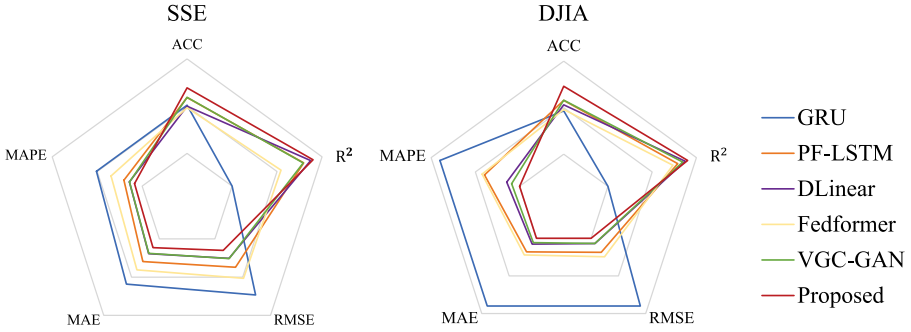


Fig. 3. Comparative graph of predictive performance between the proposed method and selected baseline methods.

Table 6
Ablation comparison. Results with **bold** are the best performance.

Method	SSE					DJIA				
	ACC	R^2	RMSE	MAE	MAPE	ACC	R^2	RMSE	MAE	MAPE
Method 1	0.6861	**0.9290**	0.1542	0.1170	0.848%	0.7277	**0.9694**	1.6771	1.2287	0.731%
Method 2	0.6849	0.9224	0.1612	0.1190	0.854%	0.7158	0.9461	1.6535	1.2485	0.719%
Method 3	**0.7038**	0.9257	0.1582	0.1172	0.850%	0.7125	0.9600	1.6230	**1.2147**	0.715%
Method 4	0.6824	0.9277	0.1507	0.1140	0.866%	0.7266	0.9681	1.6671	1.2164	0.716%
Method 5	0.6862	0.9155	0.1595	0.1190	0.894%	0.7213	0.9594	1.6540	1.2469	0.753%
Method 6	0.6543	0.9033	0.1668	0.1260	0.946%	0.7057	0.9582	1.7480	1.3552	0.805%
Method 7	0.6603	0.9027	0.1690	0.1301	0.992%	0.7025	0.9581	1.7898	1.3821	0.802%
Method 8	0.6836	0.9246	0.1500	0.1137	0.843%	0.7277	0.9619	1.6721	1.2617	0.769%
Method 9	0.6647	0.9129	0.1615	0.1254	0.92%	0.7152	0.9602	1.6653	1.2748	0.827%
Method 10	0.6795	0.9225	0.1595	0.1179	0.873%	0.7034	0.9605	1.7503	1.3207	0.779%
Proposed	0.6940	0.9226	**0.1500**	**0.1137**	**0.843%**	**0.7311**	0.9652	**1.6082**	1.2206	**0.714%**

SSA is a nonparametric time series decomposition method that dissects a given time series into the sum of its principal components. It entails constructing a trajectory matrix from the time series and employing singular value decomposition (SVD) to extract the principal components. SSA is renowned for its efficacy in extracting trend, periodic, and noise components from time series data.

Method 5 excluded the use of a sequence decomposition algorithm. Methods 6, 7, and 8 replaced the Multi-GCN with a GCN and utilized Pearson correlation plot G_P, Spearman's Rank correlation plot G_{Sp}, and DTW correlation graph G_{DTW}, respectively, instead of generating graph G_S. Method 9 eliminated the GCN and integrated a GRU model for multivariate prediction, while Method 10 adopted MSE as the loss function.

Drawing insights from Table 6, the following conclusions emerge:

(1) Comparative to Method 5, which omitted the use of a sequence decomposition algorithm, the methods incorporating decomposition algorithms exhibited improved performance. Notably, the method employing the VMD algorithm achieved the best performance. This underscores the significance of sequence decomposition algorithms in mitigating data non-stationarity and noise, consequently enhancing predictive model performance. VMD emerged as the most suitable and robust choice among decomposition algorithms.

(2) In contrast to Methods 6, 7, 8, and 9, which either employed single graphs or excluded the use of graph convolutional networks, the method employing multi-graph convolutional networks demonstrated superior performance across both datasets. This validates the effectiveness of graph convolutional networks, highlighting that multi-graph convolutional networks can capture more concealed relationships by amalgamating multiple graphs, resulting in substantial performance improvements.

(3) The proposed method outperformed Method 9, which utilized MSE as the loss function, exhibiting higher ACC and improvements in error

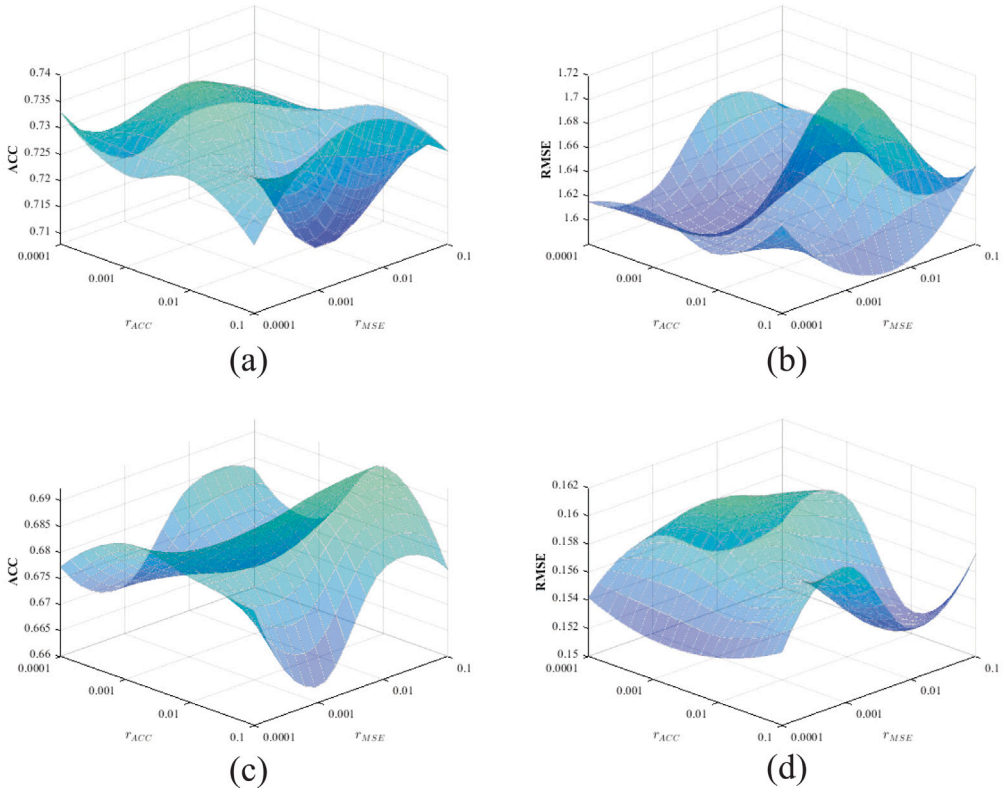


Fig. 4. Performance of different r_{MSE} and r_{ACC} on the datasets.

Table 7
Predictive performance of different sliding window lengths. Results with **bold** are the best performance.

Dataset		$L = 7$	$L = 15$	$L = 30$	$L = 45$	$L = 60$
SSE	ACC	0.6802	0.6901	**0.6940**	0.6797	0.6733
	R^2	**0.9398**	0.9289	0.9226	0.9299	0.9145
	RMSE	0.1600	0.1562	**0.1500**	0.1531	0.1550
	MAE	0.1214	0.1175	**0.1137**	0.1161	0.1144
	MAPE	0.888%	0.865%	**0.843%**	0.844%	0.867%
DJIA	ACC	0.7183	0.7256	**0.7311**	0.7160	0.7092
	R^2	0.9650	**0.9665**	0.9652	0.9646	0.9571
	RMSE	1.6895	1.6397	1.6082	**1.5671**	1.6028
	MAE	1.2885	1.2504	1.2206	**1.2024**	1.2510
	MAPE	0.747%	0.729%	**0.714%**	0.721%	0.757%

metrics such as RMSE. This underscores the effectiveness of balancing numerical precision with trend accuracy, leading to substantial enhancements in prediction performance.
(4) Each module significantly contributed to enhancing model predictive performance, with the removal of any module potentially resulting in subpar outcomes.

4.5. Analysis of hyperparameters

Table 7 illustrates the impact of varying sliding window lengths (L) on model performance. The data indicates that the model achieves its highest prediction performance when L is set to 30. Notably, if L is too small, the model's performance diminishes, likely because there is an insufficient amount of data available for the model to extract meaningful features. Conversely, if L is excessively large, the model's performance experiences a significant decline. This drop in performance may be attributed to the heightened volatility of stock data, as an extended period results in increased data noise.

Fig. 4 illustrates the impact of different regularization term coefficients, specifically r_{MSE} and r_{ACC}, on the model's performance. Subfigures (a) and (b) pertain to the model's performance on the DJIA dataset, while subfigures (c) and (d) relate to its performance on the SSE dataset. The experimental results demonstrate that an appropriate combination of these coefficients can significantly improve the model's predictive performance. However, when these coefficients are set to excessively high or excessively low values, it can have a detrimental effect on the model. Consequently, it is crucial to select appropriate regularization term coefficients based on empirical observations. In this study, a parameter combination employing an r_{MSE} of 0.01 and an r_{ACC} of 0.1 was utilized.

4.6. Trading simulation

To comprehensively assess the effectiveness of our model, we conducted an experiment employing data from October to December 2023 to simulate stock investments. Concurrently, we compared our approach with the baseline method outlined in Section 4.2.3. The estimated trading strategy operates on a daily frequency basis. Within the simulation, we established a trading strategy based on our framework's predictions. This strategy involves computing a score by aggregating the difference between the predicted closing price and the previous day's closing price, in conjunction with the accuracy of the past L_{ACC} days. Based on this score, we make investment decisions: we invest in the top k stocks with the highest scores and hold them for one day. Alternatively, we refrain from investing in a stock if the predicted value is lower than the previous day's closing price. Fig. 5 depicts the net worth curves of all models over the simulation period. As shown in Fig. 5, our proposed method consistently exhibits optimality and robustness on both datasets, with a reduction in returns observed for both datasets after transitioning to the Mean Squared Error (MSE) loss function.

Furthermore, Fig. 6 illustrates the cumulative profit curves for various alternative trading strategies. Subfigures (a) and (b) represent the trading strategy following Liu, Ma et al. (2022), which involves

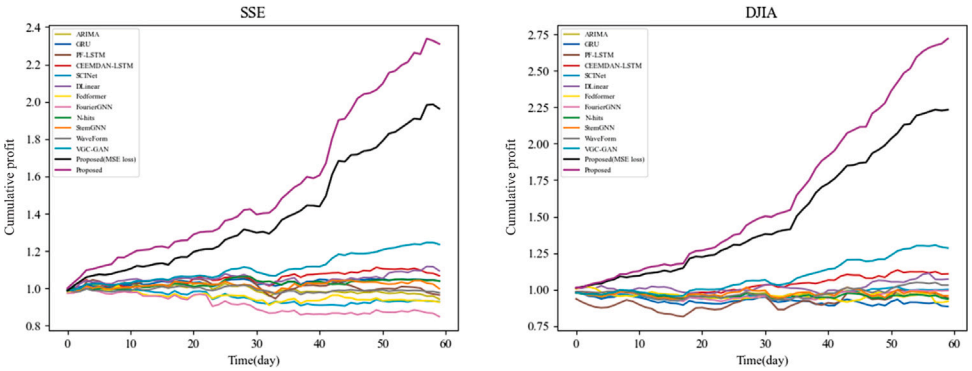


Fig. 5. Cumulative profit curves of the proposed method and baseline methods under the proposed trading strategy.

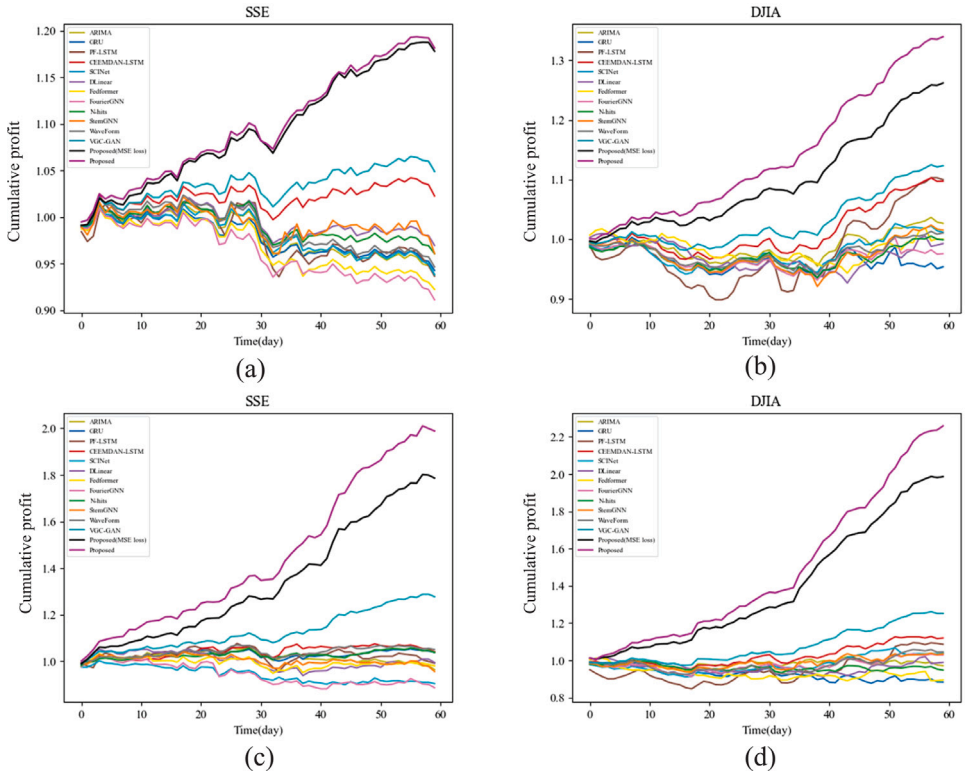


Fig. 6. Cumulative profit curves of different trading strategies.

investing, on average, in all stocks predicted to experience an increase in stock price, holding them, and selling when the predicted stock price decreases. Subfigures (c) and (d) depict a trading strategy referencing the stock recommendation system (Ma et al., 2022), where we select the top k stocks and invest in them on an average basis, holding them for one day. The results underscore the suitability of our proposed trading strategy for stock price prediction, showcasing its ability to achieve higher cumulative profits and effectively highlight performance disparities among different stock price prediction methods.

Fig. 7 depicts the influence of different values of k and L_{ACC} on cumulative profits. In practical stock market trading, diversifying investments across multiple stocks is a common strategy to mitigate risks. Specifically, when considering investment in multiple stocks with varying values of k (i.e., 3, 5, and 7), the highest return is achieved when L_{ACC} is set to 10. Smaller values of k result in fewer stocks being invested, potentially leading to higher returns, but also correspondingly increasing investment risk. Conversely, when larger values of k are chosen, the potential cumulative profits are reduced, but the associated investment risk is also diminished. Thus, to strike a balance between optimizing returns and managing risk, this study opts for setting k at 3 and L_{ACC} at 10.

5. Conclusion

This paper introduces a regularized ensemble framework with graph convolutional networks for stock price prediction. The proposed method comprises several key components. Firstly, it decomposes the original stock price sequence by utilizing Variational Mode Decomposition (VMD). This decomposition yields subsequences with distinct time scales, thereby reducing the inherent instability of stock price data and mitigating the influence of noise on model performance. Subsequently, separate relationship graphs are constructed for each of these subsequences, with hidden dependencies between nodes being learned through a multi-graph convolutional network. Following this, Gated Recurrent Units (GRUs) are employed to model the temporal dependencies of node embeddings and make predictions regarding the future closing prices of these subsequences. Finally, the predicted values for each subsequence are amalgamated to forecast the future closing price

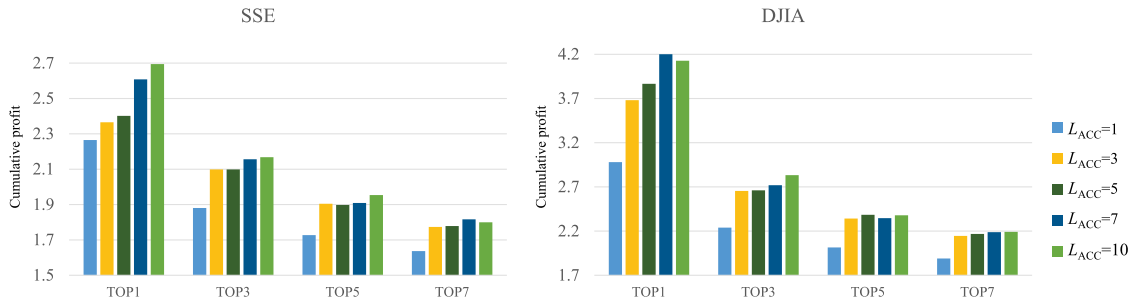


Fig. 7. The impact of different k and L_{ACC} on cumulative profit.

of the original sequence. To balance numerical precision and trend accuracy, a regularization term is introduced into the loss function. Empirical experiments conducted on real datasets demonstrate that the proposed approach yields promising predictive performance and results in higher returns in trading simulations.

In future research, we intend to further enhance our model by incorporating additional data sources such as news, tweets, and others to capture market dynamics better. Moreover, we plan to extend the applicability of this method to other time series forecasting tasks, including but not limited to traffic flow forecasting, electricity demand forecasting, and wind energy forecasting.

CRediT authorship contribution statement

Dongbo Ma: Conceptualization, Methodology, Software, Validation, Writing – original draft. **Da Yuan:** Conceptualization, Methodology, Validation, Writing – review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank all the anonymous reviewers for their helpful comments and constructive guidance. This work was supported by the National Natural Science Foundation of China under Grant 62072285. (Corresponding author: Da Yuan.)

References

Alam, M. Z., Siddikee, M. N., & Masukujjaman, M. (2013). Forecasting volatility of stock indices with ARCH model. *International Journal of Financial Research*, *4*(2), 126.

Banik, S., Sharma, N., Mangla, M., Mohanty, S. N., & Shitharth, S. (2022). LSTM based decision support system for swing trading in stock market. *Knowledge-Based Systems*, *239*, Article 107994.

Bollerslev, T., Marrone, J., Xu, L., & Zhou, H. (2014). Stock return predictability and variance risk premia: Statistical inference and international evidence. *Journal of Financial and Quantitative Analysis*, *49*(3), 633–661.

Challu, C., Olivares, K. G., Oreshkin, B. N., Garza, F., Mergenthaler-Canseco, M., & Dubrawski, A. (2022). N-hits: Neural hierarchical interpolation for time series forecasting. arXiv preprint arXiv:2201.12886.

Chen, W., Jiang, M., Zhang, W.-G., & Chen, Z. (2021). A novel graph convolutional feature based convolutional neural network for stock trend prediction. *Information Sciences*, *556*, 67–94.

Chen, Y., Wei, Z., & Huang, X. (2018). 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* (pp. 1655–1658).

Chen, Y., Wu, J., & Wu, Z. (2022). China's commercial bank stock price prediction using a novel K-means-LSTM hybrid approach. *Expert Systems with Applications*, *202*, Article 117370.

Cheng, C.-H., & Wei, L.-Y. (2014). A novel time-series model based on empirical mode decomposition for forecasting TAIEX. *Economic Modelling*, *36*, 136–141.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Cui, C., Li, X., Zhang, C., Guan, W., & Wang, M. (2023). Temporal-relational hypergraph tri-attention networks for stock trend prediction. *Pattern Recognition*, Article 109759.

Cui, C., Wang, P., Li, Y., & Zhang, Y. (2023). McVCsB: A new hybrid deep learning network for stock index prediction. *Expert Systems with Applications*, *232*, Article 120902. http://dx.doi.org/10.1016/j.eswa.2023.120902, URL https://www.sciencedirect.com/science/article/pii/S0957417423014045.

Deng, C., Huang, Y., Hasan, N., & Bao, Y. (2022). Multi-step-ahead stock price index forecasting using long short-term memory model with multivariate empirical mode decomposition. *Information Sciences*, *607*, 297–321.

Dragomiretskiy, K., & Zosso, D. (2013). Variational mode decomposition. *IEEE Transactions on Signal Processing*, *62*(3), 531–544.

Dudukcu, H. V., Taskiran, M., Taskiran, Z. G. C., & Yildirim, T. (2023). Temporal convolutional networks with RNN approach for chaotic time series prediction. *Applied Soft Computing*, *133*, Article 109945.

Feng, S., Xu, C., Zuo, Y., Chen, G., Lin, F., & XiaHou, J. (2022). Relation-aware dynamic attributed graph attention network for stocks recommendation. *Pattern Recognition*, *121*, Article 108119.

Gupta, U., Bhattacharjee, V., & Bishnu, P. S. (2022). StockNet—GRU based stock index prediction. *Expert Systems with Applications*, *207*, Article 117986.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press.

Huang, Y., Gao, Y., Gan, Y., & Ye, M. (2021). A new financial data forecasting model using genetic algorithm and long short-term memory network. *Neurocomputing*, *425*, 207–218.

Jia, P., Zhang, H., Liu, X., & Gong, X. (2021). Short-term photovoltaic power forecasting based on VMD and ISSA-GRU. *IEEE Access*, *9*, 105939–105950.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Kumar, R., Kumar, P., & Kumar, Y. (2022). Three stage fusion for effective time series forecasting using Bi-LSTM-ARIMA and improved DE-ABC algorithm. *Neural Computing and Applications*, *34*(21), 18421–18437.

Lahmiri, S. (2016). A variational mode decompoisition approach for analysis and forecasting of economic and financial time series. *Expert Systems with Applications*, *55*, 268–273.

Li, W., Bao, R., Harimoto, K., Chen, D., Xu, J., & Su, Q. (2021). Modeling the stock relation with graph network for overnight stock movement prediction. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence* (pp. 4541–4547).

Li, S., Wu, J., Jiang, X., & Xu, K. (2022). Chart GCN: Learning chart information with a graph convolutional network for stock movement prediction. *Knowledge-Based Systems*, *248*, Article 108842.

Lin, Y., Lin, Z., Liao, Y., Li, Y., Xu, J., & Yan, Y. (2022). Forecasting the realized volatility of stock price index: A hybrid model integrating CEEMDAN and LSTM. *Expert Systems with Applications*, *206*, Article 117736.

Lin, Y., Yan, Y., Xu, J., Liao, Y., & Ma, F. (2021). Forecasting stock index price using the CEEMDAN-LSTM model. *The North American Journal of Economics and Finance*, *57*, Article 101421.

Liu, T., Ma, X., Li, S., Li, X., & Zhang, C. (2022). A stock price prediction method based on meta-learning and variational mode decomposition. *Knowledge-Based Systems*, *252*, Article 109324.

Liu, T., & Shi, Y. (2022). Innovation of the component GARCH model: simulation evidence and application on the Chinese stock market. *Mathematics*, *10*(11), 1903.

Liu, M., Zeng, A., Chen, M., Xu, Z., Lai, Q., Ma, L., & Xu, Q. (2022). Scinet: time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, *35*, 5816–5828.

Lv, P., Wu, Q., Xu, J., & Shu, Y. (2022). Stock index prediction based on time series decomposition and hybrid model. *Entropy*, *24*(2), 146.

Ma, X., Karkus, P., Hsu, D., & Lee, W. S. (2020). Particle filter recurrent neural networks. *Vol. 34*, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 5101–5108).

Ma, Y., Mao, R., Lin, Q., Wu, P., & Cambria, E. (2023). Multi-source aggregated classification for stock price movement prediction. *Information Fusion*, *91*, 515–528.

Ma, D., Yuan, D., Huang, M., & Dong, L. (2024). VGC-GAN: A multi-graph convolution adversarial network for stock price prediction. *Expert Systems with Applications*, *236*, Article 121204. http://dx.doi.org/10.1016/j.eswa.2023.121204, URL https://www.sciencedirect.com/science/article/pii/S0957417423017062.

Ma, X., Zhao, T., Guo, Q., Li, X., & Zhang, C. (2022). Fuzzy hypergraph network for recommending top-K profitable stocks. *Information Sciences*, *613*, 239–255.

Park, H. J., Kim, Y., & Kim, H. Y. (2022). Stock market forecasting using a multi-task approach integrating long short-term memory and the random forest framework. *Applied Soft Computing*, *114*, Article 108106.

Qiu, Y., Song, Z., & Chen, Z. (2022). Short-term stock trends prediction based on sentiment analysis and machine learning. *Soft Computing*, *26*(5), 2209–2224.

Salvador, S., & Chan, P. (2007). Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, *11*(5), 561–580.

Si, Y.-W., & Yin, J. (2013). OBST-based segmentation approach to financial time series. *Engineering Applications of Artificial Intelligence*, *26*(10), 2581–2596.

Tang, H., Dong, P., & Shi, Y. (2019). A new approach of integrating piecewise linear representation and weighted support vector machine for forecasting stock turning points. *Applied Soft Computing*, *78*, 685–696.

Tay, F. E., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega*, *29*(4), 309–317.

Wang, J., Hu, Y., Jiang, T.-X., Tan, J., & Li, Q. (2023). Essential tensor learning for multimodal information-driven stock movement prediction. *Knowledge-Based Systems*, *262*, Article 110262.

Wang, J.-Z., Wang, J.-J., Zhang, Z.-G., & Guo, S.-P. (2011). Forecasting stock indices with back propagation neural network. *Expert Systems with Applications*, *38*(11), 14346–14355.

Xi, S., & Xu, X. (2023). Dynamic gaussian deep belief network design and stock market application. *Intelligent Data Analysis*, *27*(2), 519–534.

Xu, W., Wang, B., Liu, J., Chen, Y., Duan, P., & Hong, Z. (2022). Toward practical privacy-preserving linear regression. *Information Sciences*, *596*, 119–136.

Yang, F., Li, X., Wang, M., Zang, H., Pang, W., & Wang, M. (2023). WaveForM: graph enhanced wavelet learning for long sequence forecasting of multivariate time series. *Vol. 37*, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 10754–10761).

Ye, J., Yang, Z., Ma, M., Wang, Y., & Yang, X. (2022). ϵ-Kernel-free soft quadratic surface support vector regression. *Information Sciences*, *594*, 177–199.

Yi, K., Zhang, Q., Fan, W., He, H., Hu, L., Wang, P., An, N., Cao, L., & Niu, Z. (2023). FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective. In *Thirty-seventh conference on neural information processing systems*.

Yin, X., Yan, D., Almudaifer, A., Yan, S., & Zhou, Y. (2021). Forecasting stock prices using stock correlation graph: A graph convolutional network approach. In *2021 international joint conference on neural networks* (pp. 1–8). http://dx.doi.org/10.1109/IJCNN52387.2021.9533510.

Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are transformers effective for time series forecasting?

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R. (2022). Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning* (pp. 27268–27286). PMLR.