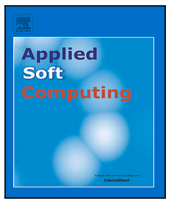
Applied Soft Computing Journal 182 (2025) 113519

Contents lists available at ScienceDirect

Applied Soft Computing

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



Model-Guided Stock Movements Prediction with Homogeneous–Heterogeneous pattern learning

Yi Zhou [a,d,e], Tai-Xiang Jiang [a,d,e], Jun Wang [b,*], Jinghua Tan [c,*]

[a] School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu, 611130, PR China
[b] School of Management Science and Engineering, Southwestern University of Finance and Economics, Chengdu, 611130, PR China
[c] College of Economics, Sichuan Agricultural University, Chengdu, 611134, PR China
[d] Kash Institute of Electronics and Information Industry, Kash, 844099, PR China
[e] Engineering Research Center of Intelligent Finance, Ministry of Education, Southwestern University of Finance and Economics, Chengdu, 611130, PR China

ARTICLE INFO

Keywords:
Unrolling network
Tensor
Stock movements

ABSTRACT

Stock movement prediction is a difficult task in the field of financial technology due to non-stationary dynamics and complex market interdependencies. Most of the existing research is based on deep neural networks, which lack interpretability. An interpretable prediction method helps uncover the mystery of the underlying operating mechanism of the securities market. In this work, we propose a model-guided method with interpretable homogeneous–heterogeneous processing for stock movement prediction. Specifically, based on that the correlations among the entities in the market are homogeneous within a short period, we unroll the iterative algorithm for solving the tensor robust principal component analysis (TRPCA) to separate the homogeneous and heterogeneous patterns from multiview data. Then, a specialized tensor-based attention for homogeneous and heterogeneous feature extraction is designed, and embedded in long short-term memory (LSTM) for better prediction. Experiments on real datasets show our model's superiority over state-of-the-art stock forecast methods.

1. Introduction

The stock market stands as one of the most influential financial markets, and predicting market movements is an extremely attractive topic for traders and investors. However, in real-world scenarios, the dynamics of the stock market is highly stochastic. It is challenging to track stock movements to avoid as much risk as possible and make optimal investment decisions. To improve prediction accuracy, machine learning [1] and deep learning techniques [2,3] are increasingly emerging in both academic research and industry.

Many factors contribute to the volatility of the stock market, and the main factors include the state of the economy, the policy environment, traders' expectations, and the mood of the online media. Therefore, it is quite difficult to predict the trend of stock volatility. Previously, scholars concentrated on studying technical indicator modality and utilizing the information obtained from feature mining for forecasting [4]. With the rise of the Internet, scholars have found that social media sentiment also has an impact on stock market volatility and have studied it as a new modality [5].

The accuracy of stock movements forecasting relies on various market information and poses a multi-modal learning challenge. Feng et al. [6] observed that integrating stock price and news information can help improve stock prediction. Cheng et al. [7] designed a graph neural network to predict financial time series, which preserves informative market information as inputs, including stock prices, news information and relations in KG. Ma et al. [8] developed a fusion model that demonstrated enhanced prediction performance by incorporating multiple sources of market information. Wang et al. [9] introduced an innovative stock correlation representation approach using the tensor format, leveraging a tensor robust principal component analysis (TRPCA) model to seamlessly integrate multi-modal and multi-temporal market data. Additionally, existing methods based on multi-modal data utilized the homogeneity of such data for prediction [10]. In the field of stock movement prediction, homogeneous patterns refer to consistent and predictable behaviors or trends observed across multiple stocks or time periods. These patterns exhibit similarities in their movement dynamics, suggesting a high degree of coherence in the market (Coherence can be well captured by the low-rankness). Conversely, heterogeneous patterns describe diverse and irregular behaviors, where stocks or time periods exhibit distinct and non-uniform movement characteristics. Heterogeneous patterns often reflect complex interactions

* Corresponding authors.
E-mail addresses: yizhou200004@gmail.com (Y. Zhou), taixiangjiang@gmail.com (T.-X. Jiang), wangjun1987@swufe.edu.cn (J. Wang), jinghuatan@sicau.edu.cn (J. Tan).

https://doi.org/10.1016/j.asoc.2025.113519
Received 19 February 2025; Received in revised form 12 June 2025; Accepted 20 June 2025
Available online 7 July 2025

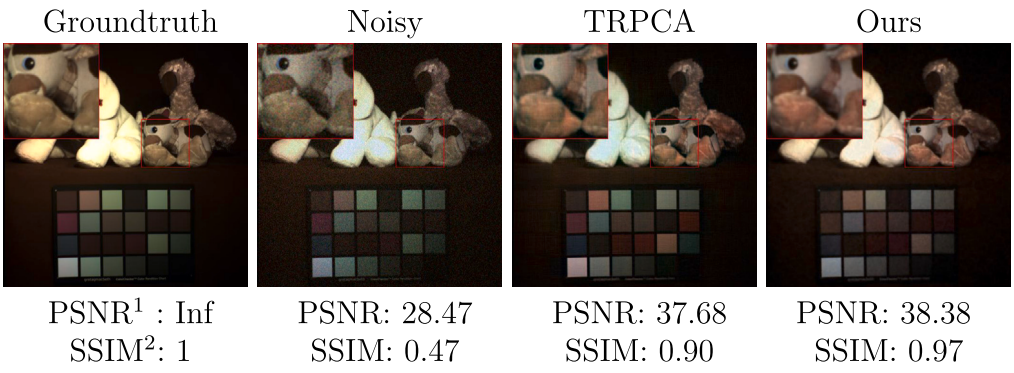


Fig. 1. Sparse noise removal results by TPRCA [19] and our TRPCA unrolling network on the multi-spectral image *stuffed toys*.[1][2]

and external influences, making them more challenging to predict. The above papers focus on the homogeneous part of the multimodal information, which is used for prediction. Each company in the market possesses idiosyncratic information, considered as heterogeneous data that complements the homogeneous information.

However, the usage of heterogeneous information is also important [11], which is viewed as complement of the homogeneous information. Therefore, the rational and effective combination of both homogeneous and heterogeneous information in multi-modal data is crucial for stock prediction.

Previous work usually splices features from different information sources into a single feature vector, while this approach ignores the interactions between various information sources [12]. To take advantage of possible correlations between different stocks, Felix et al. [13] represented the textual data through a matrix and used a sparse matrix factorization method to extract the correlation between the stock price and the textual data and used it for prediction. Ritika et al. [14] represented the NASDAQ dataset using matrices, then use principal component analysis for dimensionality reduction, and then fed the reduced data into an artificial neural network (ANN) for prediction. The natural properties of the tensor, which can represent data of arbitrary dimensions and well represent the intrinsic connections of data, make the tensor an ideal tool for processing and representing complex data [15,16]. Considering the effectiveness of tensors in fusing multidimensional data, some studies have attempted to use tensor-based approaches for modeling to preserve as much information as possible. Li et al. [17] used a tensor format to represent the raw stock data in different modes and subsequently augmented the underlying relationships inherent in the data through the tucker decomposition. Although the intrinsic connection of different modal information is considered through tensor format in [17,18], the inherent connection of different stocks is ignored, to solve this problem, Zhang et al. [11] proposed a method based on coupling matrix and tensor decomposition, which adopts the tensor format to fuse social media, historical quantitative data as a way to study their common impact on stock price movements.

While tensor-based methods are more effective and interpretable, they also suffer from some issues. Many tensor-based methods maintains two-stage, the tensor representation part, which involves tensor optimization and computations to mine the multi-linear correlations, and the prediction part, which is generally a nonlinear deep neural network. This prevents them from an end-to-end training and results in low efficiency.

In this work, we unroll a TRPCA algorithm to build the model-guided deep neural network (HHS) as the backbone of the framework. Then, a homogeneous and heterogeneous self-attention is designed to seamlessly handle the low-rank (homogeneous) and sparse (heterogeneous) patterns separated from the HHS module. Thus, our method can well handle homogeneous and heterogeneous patterns in multimodal tensor data with good effectiveness and interpretability. More specifically, we first design an attention-like correlation extraction module to extract the correlation among different companies across different modalities. Then, we design a homogeneous–heterogeneous separating network by unrolling an iterative alternating direction multiplication method (ADMM) algorithm, which is initially established for solving the TRPCA problem [19]. The network modules, including the low-rank thresholding part, the sparse thresholding part, and the multiplier updating part, are strictly consistent with the optimization algorithm. A toy example in Fig. 1 shows that an unrolling network is able to perform better than the corresponding optimization algorithm. Finally, we design an LSTM with the Homogeneous–Heterogeneous Self-Attention Mechanism (HHSAM), which comprises three key modules: Feature Map Block (FMB), Heterogeneous information Diagonal Embedding (HDE) and Homogeneous–Heterogeneous Self-attention (HHSA), to handle the low-rank (homogeneous) and sparse (heterogeneous) features separated from the TRPCA module.

The principal contributions of this research can be summarized as follows.

- With a view to extract the homogeneous and heterogeneous pattern of the multiview data, we unroll the iterative algorithm designed for TRPCA into a deep neural network. The network structure and modules are consistent with the iterative algorithm with good interpretability. Thus, homogeneous information is reliably extracted in the tensor low-rank component, while heterogeneous information is captured in the sparse component.
- Following the model-driven homogeneous–heterogeneous separation part, we present a novel homogeneous–heterogeneous attention block that can utilize company-wise heterogeneous similarity as well as homogeneous particularity, therefore yielding better prediction results.
- Experiments on the CSI 500, Nasdaq and FTSE 100 data sets indicate that our approach outperforms existing state-of-the-art methods. Investment simulations reveal that our approach attains the highest annual return rate, standing at 27.24%.

The rest of this paper is given below. Section 2 provides a review of the relevant literature on stock movement prediction and the unrolling network. Section 3 provides the basic notations and revisits the TRPCA model and the corresponding optimization algorithm. Section 4 presents the model architecture. Section 5 showcases the experimental results. Subsequently, Section 6 is dedicated to an in-depth discussion. Lastly, conclusions are provided in Section 7.

2. Related work

2.1. Stock movement prediction

The autoregressive integrated moving average (ARIMA) model [21] and the generalized autoregressive conditional heteroskedasticity

[1] https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio.
[2] Structural Similarity Index [20].

(GARCH) model [22], based on traditional econometric models, have been extensively utilized in time series analysis in the field of economics and finance. However, various commonly used prediction models and methods based on traditional statistical methods, due to the constraints of their own linear structure, have affected the further improvement of prediction accuracy for time series data, and often cannot achieve satisfactory results in practical applications.

The rise of deep learning technology allows stock prediction to better deal with these problems. Currently, deep learning-based stock prediction methods have also become one of the hot spots in stock prediction research [23,24]. Various prevalent deep learning architectures encompass convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and the Transformer model, etc. Specifically, RNN and its variant networks have demonstrated commendable performance in forecasting stock trends, as they can capture potential chaotic time series dynamics.

As a special variant of RNN, LSTM learns temporal patterns through feedback connections within the neural network, overcoming the problem of gradient disappearance or explosion caused by RNN. For instance, Li et al. [25] incorporated both quantitative indicators and news sentiments extracted from sentiment dictionaries into an LSTM-based model to predict stock prices in the Hong Kong market. In addition, recently popular transformer-based architectures [26] can handle distance dependence in sequences, but their large number of parameters and complex structure aggravate the black-box problem in deep learning, leading to difficulties in interpreting during the decision-making process.

Besides, the mentioned publications presented some shortcomings. Since the acquisition and processing of market data requires a lot of time and resources, the quality and stability of the data also need to be guaranteed. The black-box nature of deep learning models makes its prediction outcomes difficult to interpret and understand, which poses certain challenges to research. Essentially, the uniqueness of financial trading requires robust predictive results and revealing mechanisms of stock movements, rather than just a winning accuracy number.

2.2. Model-driven learning via unrolling

The application of algorithm unrolling – a methodology that converts iterative optimization processes into interpretable neural architectures – remains an emerging frontier in stock price forecasting. Unlike conventional deep learning models, algorithm unrolling explicitly retains mathematical connections to classical financial theories while leveraging data-driven learning capabilities. This hybrid approach systematically unfolds iterative algorithms (e.g., gradient-based optimizers, sparse recovery methods) into layered networks, enabling end-to-end training with built-in domain constraints.

As early as 2010, Gregor and LeCun et al. [27] were inspired by the iterative shrinkage and thresholding algorithm (ISTA) and proposed the first deep expansion-based framework, LISTA, which expanded the ISTA algorithm into a non-linear sparse coding feedforward network. Based on LISTA, scholars have proposed some image super-resolution networks based on sparse coding [28]. In recent years, depth-unfolding strategies have flourished in the field of image processing, including front-and-back background separation, image denoising, and image enhancement. For example, Yang et al. [29] expanded the iterative process of the ADMM algorithm into a novel deep network and achieved superior performance in the MRI compressed sensing task. In the literature, Zhang et al. [30] introduced ISTA-Net, a deep network built upon the ISTA expansion, using nonlinear transformation to solve the proximal mapping related to the sparsity-induced regularizer. A large number of experiments have shown that ISTA-Net advances in compressed sensing reconstruction.

In financial contexts, unrolling offers unique advantages for modeling non-stationary market dynamics, particularly in terms of interpretability. Its modular design allows for the direct integration of volatility-aware priors into network layers, ensuring that the model's decisions are grounded in interpretable financial principles. Early studies demonstrate its potential in multi-scale feature extraction from noisy market data while maintaining computational tractability. Although still underexplored compared to mainstream black-box models, unrolling frameworks show promise in balancing performance with traceable reasoning, a critical consideration for risk-sensitive financial applications.

Interpretability remains a critical challenge in stock prediction research, as current deep learning approaches that demonstrate superior predictive performance typically exhibit opaque "black-box" characteristics. While several studies have attempted to address this issue through different methodological perspectives, comprehensive model interpretability has yet to be achieved. Previous work has made progress in understanding different aspects of the problem, Hu et al. [31] conducted empirical analyses of attention weights within textual corpora to evaluate the relative importance of financial news articles. Subsequent work by Dang et al. [32] developed a multimodal neural architecture designed to filter news content relevant for stock market forecasting. More recently, Li et al. [33] proposed an innovative Prediction-Explanation Network (PEN) that aligns textual information with pricing data streams through joint representation learning. The PEN framework employs a salient vector mechanism to capture text-price correlations, enabling identification of potentially influential news content that can subsequently provide explanatory rationales for observed price movements.

Overall, the unrolling strategy combines good performance with a high degree of interpretability, which makes it not only capable of achieving efficient and accurate results when dealing with complex problems, but also provides researchers and developers with clear ideas and a basis for decision-making.

3. Revisiting tensor robust principal component analysis

First, we summarize the core tensor algebra conventions essential to our methodology. Throughout this paper, we use $\mathcal{X}$ to denote a tensor, $\mathbf{X}$ to denote a matrix and $\boldsymbol{X}$ to denote a vector. Key operational constructs—including the tensor-tensor product (t-product) and tensor nuclear norm($\|\cdot\|_{\mathrm{TNN}}$) are contextually introduced here to guide the technical narrative, with their formal mathematical definitions rigorously derived in Appendix A. The t-SVD framework, the mathematical foundation of our approach, originates from [34,35] and has received tremendous attention in recent years. Theoretical guarantees for separating the low-rank tensor from sparse corruptions, namely the tensor robust principal component analysis (TRPCA), are established in [19]. Wang et al. combining the TRPCA model with an attention-based LSTM for stock movements prediction [9].

Lu et al. [19] proposed a tensor robust principal component analysis (TRPCA) model based on the tensor nuclear norm (TNN) derived from the t-SVD framework, aiming to exactly separate the low-rank tensor structure and sparse outliers from high-dimensional data. Given a tensor $\mathcal{X}$, the formulation of the model is as follows.

$$\begin{aligned} &\min_{\mathcal{L},\mathcal{T}} \ \|\mathcal{L}\|_{\mathrm{TNN}} + \lambda\|\mathcal{T}\|_1 \\ &\text{s.t.} \quad \mathcal{X} = \mathcal{L} + \mathcal{T} \end{aligned} \tag{1}$$

where $\mathcal{L}$ and $\mathcal{T}$, respectively, represent the low-rank component and the sparse component, λ represents a non-negative parameter, $\|\cdot\|_1$ denotes the ℓ_1 norm, and $\|\cdot\|_{\mathrm{TNN}}$ denotes the tensor nuclear norm.

The above problem (1) can be efficiently solved using the ADMM algorithm [36], which decomposes the original problem into tractable sub-problems while ensuring constraint satisfaction. First, we construct the augmented Lagrangian function:

$$L_\beta(\mathcal{L},\mathcal{T},\mathcal{M}) = \|\mathcal{L}\|_{\mathrm{TNN}} + \lambda\|\mathcal{T}\|_1 + \langle\mathcal{M},\mathcal{X}-\mathcal{L}-\mathcal{T}\rangle + \frac{\beta}{2}\|\mathcal{X}-\mathcal{L}-\mathcal{T}\|_F^2, \tag{2}$$

where $\mathcal{M}$ denotes the Lagrange multiplier, and $\beta > 0$ is the penalty parameter.

After that, the ADMM alternately updates $\{\mathcal{L}, \mathcal{T}\mathcal{M}\}$ as follows.

$$\begin{cases} \mathcal{L}^{(k+1)} = \arg\min_{\mathcal{L}} \|\mathcal{L}\|_{TNN} + \frac{\beta}{2}\left\|\mathcal{X} - \mathcal{L} - \mathcal{T}^{(k)} + \frac{\mathcal{M}^{(k)}}{\beta}\right\|_F^2 & (3) \\ \mathcal{T}^{(k+1)} = \arg\min_{\mathcal{T}} \lambda\|\mathcal{T}\|_1 + \frac{\beta}{2}\left\|\mathcal{X} - \mathcal{L}^{(k+1)} - \mathcal{T} + \frac{\mathcal{M}^{(k)}}{\beta}\right\|_F^2 & (4) \\ \mathcal{M}^{(k+1)} = \mathcal{M}^{(k)} + \beta\left(\mathcal{X} - \mathcal{L}^{(k+1)} - \mathcal{T}^{(k+1)}\right) & (5) \end{cases}$$

The complete implementation workflow, including initialization and termination criteria, is formally described in Algorithm 1. The detailed explanation of Algorithm 1 can be seen in Appendix B.

Algorithm 1 ADMM iterations for tensor RPCA

Input: The observed tensor $\mathcal{X}, \mathcal{L}, \mathcal{T}, \mathcal{M}$

Initialization: The error threshold $\eta_{\mathcal{L}}$, $\eta_{\mathcal{T}}$; the Lagrangian parameter β; the non-negative parameter λ.

while $\eta_{\mathcal{L}} > 1 \times 10^{-6}$ **and** $\eta_{\mathcal{T}} > 1 \times 10^{-6}$ **do**
  Decompose the tensor $\mathcal{L}^k$ into $\mathcal{U} * \mathcal{S} * \mathcal{V}^H$ via t-SVD
  Update $\mathcal{L}^{k+1}$ via $\mathcal{L}^k = \mathcal{U} * \text{Shrink}_{\frac{1}{\beta}}(\mathcal{S}) * \mathcal{V}^H$;
  Update $\mathcal{T}^{k+1}$ via $\mathcal{T}^{k+1} = \text{Shrink}_{\frac{\lambda}{\beta}}(\mathcal{X} - \mathcal{L}^{k+1} + \frac{\mathcal{M}^k}{\beta})$;
  Update $\mathcal{M}^{k+1}$ via $\mathcal{M}^{k+1} = \mathcal{M}^k + \beta(\mathcal{X} - \mathcal{L}^{k+1} - \mathcal{T}^{k+1})$
  Update $\eta_{\mathcal{L}} = \frac{\|\mathcal{L}^{k+1} - \mathcal{L}^k + 1\times10^{-10}\|_2}{\|\mathcal{L}^{k-1} + 1\times10^{-10}\|_2}$
  Update $\eta_{\mathcal{T}} = \frac{\|\mathcal{T}^{k+1} - \mathcal{T}^k + 1\times10^{-10}\|_2}{\|\mathcal{T}^{k-1} + 1\times10^{-10}\|_2}$
end while

Output: The tensor $\mathcal{L}^{k+1}, \mathcal{T}^{k+1}, \mathcal{M}^{k+1}$.

Theorem 4.1 in [19] provides a mathematical guarantee that the low-rank component and the sparse component can be exactly separated into $\mathcal{L}$ and $\mathcal{T}$, respectively, under certain conditions, via Algorithm 1. The basic assumption in [19] is that the low-rank part satisfies the tensor incoherent condition. Generally speaking, this assumption holds when the low-rank part is not sparse and the sparse part is not low-rank.

Based on this theoretical foundation, Wang et al. [9] presents systematic empirical evidence demonstrating that higher-order tensor representations constructed from multidimensional financial market datasets exhibit prominent low-rank structural characteristics. By applying the TRPCA framework to analyze these financial tensors, [9] reveals that the original tensor can be effectively decomposed into a superposition structure comprising low-rank principal components and sparse noise elements.

This theoretical–empirical synergy not only reinforces TRPCA's methodological robustness but also substantiates its applicability in processing complex financial datasets.

4. Model-guided stock movements prediction with homogeneous–heterogeneous processing

Fig. 2 is an overview of our model. First, the Attention-like Homogeneous Correlation Extraction Module (AHCE) extracts homogeneous information from the original tensor for subsequent fusion. Then the model-guided deep neural network (HHS) (i.e., the expanded network based on the TRPCA algorithm) based on the TRPCA algorithm divides the original stock data into homogeneous pattern and heterogeneous pattern. The initially separated homogeneity and heterogeneity patterns are further utilized by the homogeneous–heterogeneous self-attention mechanism (HHSAM) to generate features that represent stock prediction information.

4.1. Attention-like homogeneous correlation extraction module

In this section, we integrate multimodal stock market information into a tensor format, preserving their internal structure. In the securities market, the relationship computed along different modes can be viewed as a projection of the true relationship of the individual firms within the market in different modalities. At the same time, for a short period of time, we can view this relationship as homogeneous [9].

Consider a market scenario where there are N stocks being traded over a time frame of T consecutive trading days, each stock has m information modalities (i.e., time mode and basic mode), and each mode has d_m features (such as opening price and closing price). We represent the data of each stock in certain mode every day with a vector $\boldsymbol{x}$ whose dimension is $d \times 1$, where d represents the number of features in this mode. That is, all stock information under this mode can be represented by a matrix $\mathbf{A}$ of size $N \times d$. Along the first dimension of $\mathbf{A}$, the sum of the outer products of the features of the current stock and the other stocks is computed, expanded to a one-dimensional array, and used as the i_{th} row of the matrix $\mathbf{B}$. The resulting matrix $\mathbf{B}$ is of size $N \times d^2$. With the outer product operation described above, points in the original stock feature space can be mapped to a new feature space. In this new space, each dimension represents a combinatorial interaction between the original features of different stocks. An initial similarity matrix $\mathbf{C} = \mathbf{B} \cdot \mathbf{B}^H$ can be approximated by computing the dot product of each row (representing the features of a stock) and each column (also representing the features of a stock) in $\mathbf{B}$. Therefore, the information under all m modes forms a tensor $\mathcal{C}$ of size $(m \times T, N, N)$. The above is the original data processing process.

In a short period of time, the relationships between different companies can be regarded as homogeneous, so the homogeneous correlations can be extracted first. In the attention mechanism, the dot product $\mathbf{Q} \cdot \mathbf{K}^T$ measures the similarity between the query vector and the key vector in the feature space, which is similar to the correlation coefficient used in statistics to measure the correlation between two variables. Inspired by this, we introduce a novel attention-like mechanism to model both intra-modality information and inter-modality interconnectivity. By modeling both intra-modality and intermodality relations, we aim to capture the joint effects of input modalities while preserving modality-specific features. The computation of the attention-like value is detailed below:

$$\mathbf{Q}_C = \mathbf{W}_Q \cdot \mathcal{C}^{(i)}, \quad (6)$$

$$\mathbf{K}_C = \mathbf{W}_K \cdot \mathcal{C}^{(i)}, \quad (7)$$

$$\mathcal{X}^{(i)} = \text{Attn}(\mathbf{Q}_C, \mathbf{K}_C) = \frac{\mathbf{Q}_C \cdot \mathbf{K}_C^T}{\lambda}, \quad (8)$$

where λ is a learnable parameter with an initial value of 1, $i = 1, \ldots, n_1$.

Considering the homogeneity between companies in a short period of time, AHCE is used to capture the homogeneity in the initial data. In this module, we believe that the homogeneous pattern obtained is not "pure", that is, it contains a small amount of heterogeneous information, which also provides conditions for the subsequent separation of homogeneous and heterogeneous patterns.

4.2. Model-guided separation of homogeneous and heterogeneous pattern module

In the context of predicting the trend of fluctuation in the stock market, we believe that the stock data have low-rank characteristics. This inspires us to use a lower-dimensional approach to effectively approximate the representation of data and use fewer features to describe most of the fluctuations in the original stock data without considering too much redundant information, thus improving the efficiency of prediction.

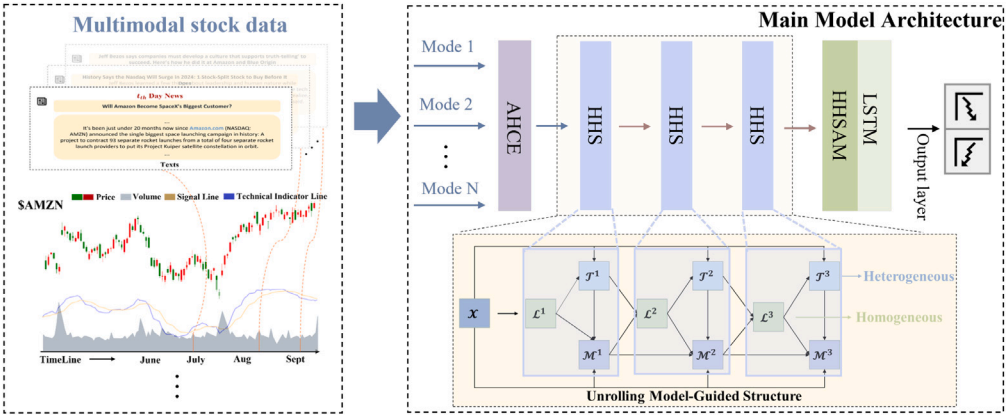


Fig. 2. The overview of the proposed model, including tensor-based feature extraction module (AHCE), model-guided separation of homogeneous–heterogeneous pattern module (HHS) and homogeneous–heterogeneous self-attention mechanism (HHSAM).

From a technical standpoint, homogeneous patterns can be associated with the low-rank component in the context of TRPCA, which captures the underlying stable and coherent trends in the data. On the other hand, heterogeneous patterns describe diverse and irregular behaviors, where stocks or time periods exhibit distinct and non-uniform movement characteristics. These heterogeneous patterns often reflect complex interactions and external influences, making them more challenging to predict. In TRPCA, these heterogeneous patterns can be attributed to the noise component, which represents the irregular and unpredictable fluctuations in the data. The separation of these two components through TRPCA provides a powerful framework for analyzing and predicting stock movements by distinguishing between stable trends and volatile noises.

Assuming that the homogeneity pattern records obtained through the AHCE module are $\mathcal{X}$, it can be decomposed into $\mathcal{X} = \mathcal{L} + \mathcal{T}$, where $\mathcal{L}$ is the low-rank component and $\mathcal{T}$ is the sparse component. In order to better separate homogeneous pattern and heterogeneous pattern of stocks, we use the tensor principal component analysis model as shown in (1) to decompose.

The above problem (1) can be solved using the ADMM algorithm, but it will take at least dozens of iterations to obtain a satisfactory solution. Meanwhile, setting hyperparameters, such as λ, is challenging. Therefore, in this study, instead of using a fixed paradigm for iterative solutions by learning from the data itself, i.e., by constructing interpretable networks (HHS) instead of the original numerical iterative solution.

In order to design the HHS, we start by mapping the ADMM iteration process onto a data flow graph. As illustrated in the upper middle part of Fig. 3, the graph comprises nodes representing the three operations of the ADMM algorithm and directed edges indicating the data flow between these operations. These operations can be summarized as low-rank layers, sparse layers, and multiplier update layers. The general framework is built by sequentially connecting each layer. Next, we will discuss them in details.

Low-rank Layer ($\mathcal{L}^k$): Before performing tensor decomposition, we first perform the Fourier transform on the data. The essence of the Fourier transform is a linear transformation, so we use a fully connected layer instead of the Fourier transform. By the definition of t-SVD, it follows that the kth frontal slice of the transformed tensor $\mathcal{X}^{(k)} = \mathcal{U}^{(k)}\mathcal{S}^{(k)}\mathcal{V}^{(k)H}$. Furthermore, because the minimization of is equivalent to the minimization of $\sum_{k=1}^{n_3} \|\mathbf{U}^{(k)H}\tilde{f}(\mathcal{X})^{(k)}\mathbf{V}^{(k)}\|_1$. The unitary matrices $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$ ($k = 1, 2, \ldots, n_3$) based on t-SVD are directly learned from the data, rather than relying on fixed paradigms. They correspond to the row and column operations respectively. To improve the low-rank representation of the transformed tensors, we utilize a multilayer convolutional neural network with nonlinear characteristics to encapsulate the transformation, enhancing the expressiveness of the neural network-driven process. Specifically, the motivations involve two points: (i) the convolution operation can effectively replace the row and column operations. (ii) Following the t-SVD transformation, the intricate features associated with small singular values are not reliably preserved. Taking into account the factors mentioned above, we use convolution instead of $\mathbf{U}^H$ and $\mathbf{V}$. After the transformation, the system maintains its ability to adapt to data and effectively extract features.

The low-rank layer structure designed according to the above process is depicted in Fig. 3. In line with the traditional TNN approach, we start by employing the neural network in mode-3 to investigate the connections between frontal sections. Likewise, the inverse transform is performed along mode-3. Meanwhile, to describe the non-linear connection between the first and second dimensions, we stack ConRCon layers (i.e. energy concentration). By employing nonlinear modeling within the low-rank layer, minimizing the TRPCA-based TNN enables the acquisition of a lower-rank tensor $\mathcal{L}$, thereby leading to a superior low-rank representation.

Sparse Layer ($\mathcal{T}^k$): The contraction operator is essentially a nonlinear function, and this layer uses the ReLu function instead of the sparse term update process formula, and the output of this layer at stage kth is specified as: $\mathcal{T}^k = \max\left(\frac{\lambda}{\beta}, \mathcal{X} - \mathcal{L}^k + \frac{\mathcal{M}^{k-1}}{\beta}\right)$, where β, λ are all learnable parameters with an initial value of 0.01.

Multiplier Update Layer ($\mathcal{M}^k$): This layer is defined by the Lagrange multiplier update process. The output of this layer at stage k is expressed as $\mathcal{M}^k = \mathcal{M}^{k-1} + \beta(\mathcal{X} - \mathcal{L}^k - \mathcal{T}^k)$, the network designation of this layer strictly follows the formula.

The iterative solution process of our proposed modules and algorithms is highly consistent and highly interpretable. The output of the HHS is the low-rank component and the sparse component of the original data, which represent homogeneous pattern and heterogeneous pattern, respectively. Next, we consider further extracting the predictive features in the homogeneous and heterogeneous patterns.

4.3. Homogeneous–heterogeneous-self-attention for prediction

In Section 4.2, the raw data are separated into homogeneous and heterogeneous patterns through the HHS module. In previous studies, scholars have focused on the market information embedded in the homogeneous pattern while ignoring the heterogeneous pattern. Therefore, we propose the HHSAM to further capture the special information in the heterogeneous pattern, which can improve the performance of the forecasting task. Specifically, the FMB module models firm homogeneity by calculating homogeneity weights along the firm dimension. At the same time, the HDE module calculates specificity weights that contain heterogeneous information to model company heterogeneity. By integrating these two sets of weights and applying them to company feature maps, the proposed mechanism can effectively capture both the

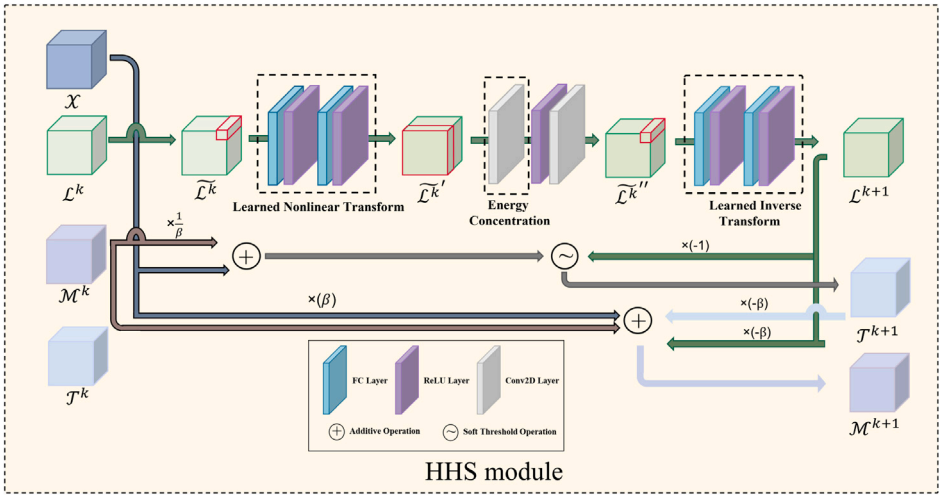


Fig. 3. Details of HHS.

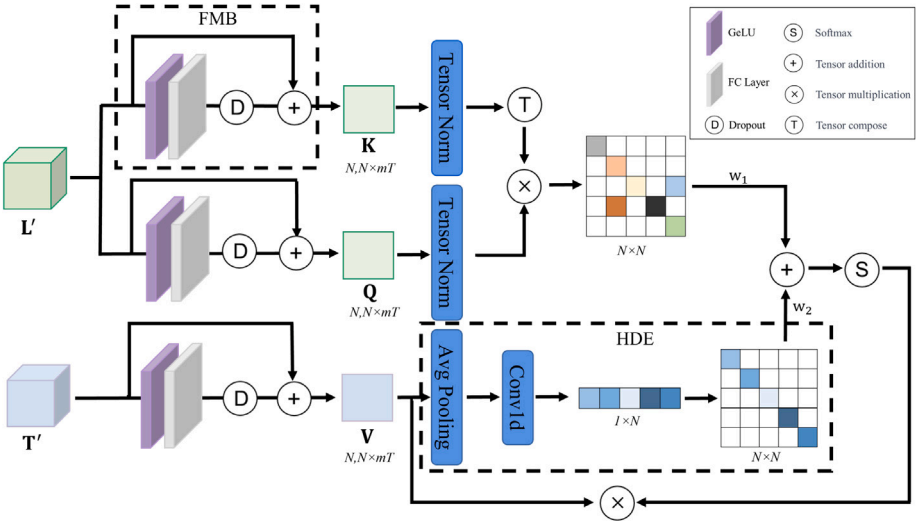


Fig. 4. Overview of homogeneous–heterogeneous-self-attention.

similarities and unique characteristics of companies. This enables the model to accurately represent company-wise correlation.

The tensor $\mathcal{L}$ and $\mathcal{T}$ obtained from the HHS module are subjected to reshaping operations to obtain a matrix of size $(N, N \times mT)$, and then they are used as inputs to this module, which are denoted as $\mathbf{L}'$, $\mathbf{T}'$ respectively. In traditional self-attention mechanisms, the size of the $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are usually the same, where $\mathbf{Q} = \mathbf{W}_q\mathbf{L}'$, $\mathbf{K} = \mathbf{W}_k\mathbf{L}'$, $\mathbf{V} = \mathbf{W}_v\mathbf{T}'$. Unlike previous studies, the query matrix, key matrix, and value matrix here are obtained through the FMB module that we designed. As shown in Fig. 4, FMB consists of two layers of activation function and linear, which linearly maps the extracted homogeneity and heterogeneity pattern to the hidden space to avoid information mixing. Company-wise Attention captures the correlation along the company dimension. Both $\mathbf{Q}$ and $\mathbf{K}$ are obtained by linear mapping of homogeneous pattern. They measure the similarity between each stock (i.e., the similarity of the company dimension). Therefore, there are $N \times N$ similar weights, and the weights on the diagonal represent the company itself. In practical implementations, $\mathbf{K}$ and $\mathbf{Q}$ are normalized along the corporate dimension with tensor norm and multiplied by a weight factor to enhance the representational power. $\mathbf{V}$ is obtained by linear mapping the heterogeneity information, measuring the difference between each stock, so we embed the $1 \times N$ specificity weights into a diagonal matrix to obtain $N \times N$ specificity.

Furthermore, to further capture the predictive properties in heterogeneity, we propose an extraction mechanism called HDE for heterogeneity information. Previous research did not consider the potential correlation between heterogeneous information. By comprehensively analyzing this information, the driving factors of market changes can be discovered. In order to fill this gap, the $\mathbf{V}$ obtained by FMB is used to obtain the heterogeneity characteristics of different company dimensions through global average pooling, and then $1d$ convolution is fully connected along the company dimension, which can effectively obtain the upgraded version of heterogeneity advanced predictive characteristics. Therefore, we propose the Homogeneous–Heterogeneous Self-attention as

$$\mathrm{HHSA}(\mathcal{L}, \mathcal{T}) = \mathtt{Softmax}(w_1 \frac{\mathbf{Q}\mathbf{K}^T}{\|\mathbf{Q}\| \cdot \|\mathbf{K}\|} + w_2(\mathrm{HDE}(\mathbf{V}))). \tag{9}$$

The self-attention mechanism in (9) is a key component to integrate the interpretable low-rank component $\mathcal{L}$ and the sparse component $\mathcal{T}$ in the unrolling output, which is consistent with the theoretical underpinnings in the relevant literature. It has been shown that TR-PCA can effectively capture this low-rank and sparse structure [19], and previous empirical studies have demonstrated that decomposing financial data into low-rank and sparse components can enhance model performance [9]. Integrating homogeneous and heterogeneous components into the self-attention mechanism further extends this proven

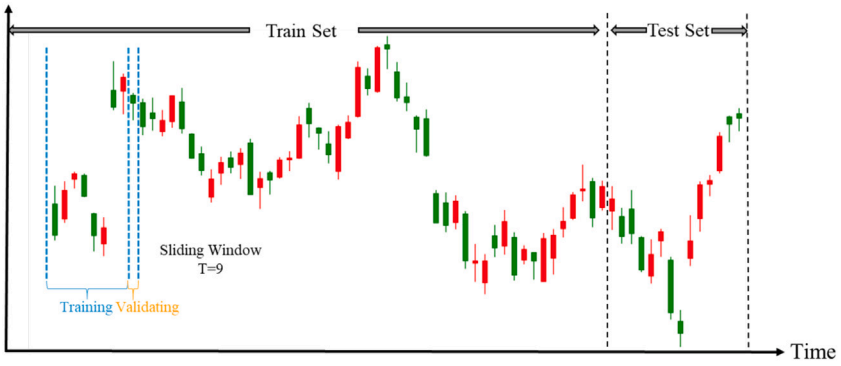


Fig. 5. Data partition diagram.

approach. The HHSA combines the advantages of the TRPCA unrolling with the flexibility of the self-attention mechanism, allowing the model to account for both low-rank trends and sparse anomalies while still maintaining a degree of interpretability.

Finally, we integrate the HHSA mechanism and LSTM for prediction. Specifically, firstly, homogeneous and heterogeneous pattern is processed through the HHSA module to capture the relationships between homogeneous–homogeneous and homogeneous–heterogeneous while encoding these relationships through weights. The output of the HHSA module is then reshaped to fit the input format of the LSTM. Next, the LSTM layer uses these encoded input sequences to learn the dependencies between time steps. Finally, the output of the LSTM layer is processed through a fully connected layer and a Sigmoid activation function to generate predictions. This combination exploits the ability of the HHSA mechanism in capturing the complex relationships of homogeneous–heterogeneous information and the strength of the LSTM in processing time-series data, which improves the model's prediction accuracy and generalization to securities market data.

5. Experimental evaluation

In this chapter, we first carry out experiments with real stock market data and contrast our model with leading stock prediction frameworks. In addition, through the ablation study, we validate the effectiveness of the primary modules of our model.

5.1. Data set and experimental settings

The experiment is conducted on the daily stock transaction data of CSI 500, Nasdaq, CMIN-US, and FTSE 100. The data spans:

- CSI 500: January 1 to December 31, 2013;
- Nasdaq: January 1, 2020 to March 31, 2021;
- CMIN-US[3]: January 2, 2018 to December 31, 2021;
- FTSE 100[4]: January 7, 2014 to June 30, 2018.

The CSI 500 data reflects China's recovery after the financial crisis. After three years of bear market, the market showed signs of easing its decline and gradually stabilizing in 2013. The Nasdaq data covers the COVID-19 pandemic period, including the market crash in $Q1$ 2020 and subsequent tech stock rally. The CMIN-US data shows the different dynamics of stock trends before and during the epidemic, and the FTSE 100 data incorporates Brexit uncertainties and post-pandemic market adjustments in the UK.

It should be noted that the CSI 500 data is limited to listed companies in mainland China, excluding Hong Kong and overseas listed companies. Nasdaq data does not include non-US companies and IPOs before 2020 to control survivor bias. FTSE 100 data excludes companies delisted during the UK's post-Brexit regulatory overhaul and lacks news sentiment data due to limited public news API coverage for UK markets. The transaction data is obtained from the Center for Research in Security Prices. The fundamental data consists of the stock prices (such as opening price), volume, turnover rate, price-to-earnings ratio, and price-to-book ratio. The news within our dataset is gathered through web crawling from East Money[5] and Bloomberg.[6] News collection is unavailable for FTSE 100; however, its technical indicators mitigate this gap by capturing short-term market dynamics. News collection is restricted to trading days within the selected time frame.

In terms of data partitioning, we use a "sliding window" to intercept fragments of the original sequence, thereby reshaping the original data into samples of a specified length for modeling. The sliding window T is a hyperparameter. Taking the T-value equal to 9 as an example, specifically, the division of the data set can be shown in Fig. 5. As shown in Fig. 6, we test the results of our method under different T values on two data sets. On the CSI 500 dataset, accuracy is highest when $T = 13$, and on the Nasdaq, FTSE 100 and CMIN-US datasets, accuracy is highest when $T = 9$, $T = 7$ and $T = 5$, respectively. During training, we use the Adam optimizer, and the batch size is set as 16. To prevent overfitting, we use a dynamic learning rate adjustment strategy. The initial learning rate is set to 0.001, the step size is set to 100 and the learning rate is gradually reduced in the later stages of training to make the model converge more smoothly to the optimal solution.

5.2. Evaluation indicators

In long-term studies, the forecasting of stock price movements is approached as a binary classification problem. Therefore, building upon prior research methodologies, the prediction performance is evaluated by four indicators: accuracy (Acc) and precision (Prec), F1-score and recall, which are hereby delineated.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{10}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{11}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{12}$$

$$\text{F1-score} = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{13}$$

where TP, TN, FP, and FN represent the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), respectively.

[3] https://github.com/BigRoddy/CMIN-Dataset.
[4] https://datalab.snu.ac.kr/dtml/.
[5] https://www.eastmoney.com/.
[6] https://www.bloombergneweconomy.com/.

(a) CSI500 data set (b) Nasdaq data set

(c) TFSE100 data set (d) CMIN-US data set

Fig. 6. Evaluation metrics for different sliding windows.

Table 1
Evaluation metrics for the prediction outcomes of various approaches on the CSI 500 set. The **top** and second highest values are respectively emphasized using **bold text** and underlining.

Methods	ACC	Precision	F1-score	Recall
GAT [38]	0.5261	0.5013	0.5064	0.5145
FinGAT [39]	0.5262	0.5085	0.5155	0.5277
AD-GAT [40]	0.5234	0.5007	0.5114	0.5215
SPLSTM [25]	0.5877	0.5202	0.5519	0.5910
TLSTM [9]	0.5746	0.5292	0.5572	0.5996
CLSR [37]	0.5295	0.4983	0.5031	0.5103
OURS	**0.6040**	**0.5454**	**0.5876**	**0.6231**

Table 2
Evaluation metrics for the prediction outcomes of various approaches on the Nasdaq data set. The **top** and second highest values are respectively emphasized using **bold text** and underlining.

Methods	ACC	Precision	F1-score	Recall
GAT [38]	0.5170	0.5082	0.5246	0.5494
FinGAT [39]	0.5051	0.5031	0.5232	0.5569
AD-GAT [40]	0.5196	0.5168	0.5306	0.5520
SPLSTM [25]	0.5403	0.5232	0.5712	0.6275
TLSTM [9]	0.5395	0.5241	0.5850	0.6489
CLSR [37]	0.5257	0.5089	0.5288	0.5496
OURS	**0.5481**	**0.5469**	**0.5993**	**0.6690**

Table 3
Evaluation metrics for the prediction outcomes of various approaches on the FTSE 100 data set. The **top** and second highest values are respectively emphasized using **bold text** and underlining.

Methods	ACC	Precision	F1-score	Recall
GAT [38]	0.5017	0.5048	0.5069	0.5098
FinGAT [39]	0.5004	0.5007	0.5023	0.5046
AD-GAT [40]	0.5028	0.5015	0.5037	0.5070
SPLSTM [25]	0.5159	0.5155	0.5122	0.5164
TLSTM [9]	0.5177	0.5175	0.5159	0.5225
CLSR [37]	0.5092	0.5003	0.5039	0.5102
OURS	**0.5267**	**0.5272**	**0.5342**	**0.5437**

5.3. Comparisons

We compare the our model with six widely-applied sequential models, i.e., the sentiment-price LSTM (SPLSTM) [25], the tensor based LSTM (TLSTM) [9], the contrastive learning framework for stock representations (CLSR) [37], the graph attention networks (GAT) [38], the financial graph attention networks (FinGAT) [39] and the attribute-driven graph attention networks (AD-GAT) [40].

- SPLSTM [25]: it utilizes the LSTM architecture to integrate technical stock price indicators with news sentiment from textual news articles for stock price prediction.
- TLSTM [9]: it combines a tensor representation and fusion framework with the LSTM framework based on attention mechanism.
- CLSR [37]: it leverages comparative learning in conjunction with the Informer framework for stock forecasting. This framework is adept at capturing both global and local patterns in extended time series data. Additionally, contrastive learning is implemented to effectively uncover potential relationships between samples and mitigate data uncertainty.
- GAT [38]: it is a powerful graph neural network structure that can flexibly capture the interrelationships between nodes in a graph by introducing an attention mechanism.
- FinGAT [39]: it combines the LSTM with a variant of GAT with a two-stage attention mechanism.
- AD-GAT [40]: it introduces a pioneering attribute-driven graph attention network, which is crafted to capture the attribute-sensitive momentum overflow effect and deduce latent relationships among listed corporations, ultimately elevating prediction accuracy.

Tables 1 and 2 show the prediction performance of the different methods on the CSI 500 and Nasdaq data sets, respectively. Overall, our proposed method achieves the best performance on all evaluation metrics on both datasets. On the CSI 500 dataset, our proposed method achieves 0.6040 for predicted ACC (at least 5.94%), which is a significant improvement compared to all other methods. On the Nasdaq dataset, our method also shows a balanced performance across all metrics. It achieves the highest Recall and F1-score, while maintaining strong performance in ACC and precision.

Tables 3 and 4 show the prediction performance of the different methods on the FTSE 100 and CMIN-US datasets, respectively. Overall, our proposed method achieves the best performance on all evaluation metrics on both datasets. On the FTSE 100 dataset, our proposed method achieves 0.5267, 0.5272, 0.5342 and 0.5437 for predicted ACC, Precision, F1-score and Recall, respectively, which is a significant improvement compared to all other methods. On the CMIN-US

Table 4
Evaluation metrics for the prediction outcomes of various approaches on the CMIN-US data set. The **top** and second highest values are respectively emphasized using **bold text** and underlining.

Methods	ACC	Precision	F1-score	Recall
GAT [38]	0.5178	0.5112	0.5471	0.5871
FinGAT [39]	0.5113	0.5088	0.5263	0.5356
AD-GAT [40]	0.5213	0.5246	0.5563	0.5981
SPLSTM [25]	0.5329	0.5496	0.5838	0.6319
TLSTM [9]	0.5343	0.5557	0.5891	0.6354
CLSR [37]	0.5209	0.5092	0.5170	0.5281
OURS	**0.5463**	**0.5609**	**0.6065**	**0.6703**

Table 5
Results of statistical tests of the results of the various methods of prediction.

Model	CSI 500		Nasdaq		FTSE 100		CMIN-US	
	χ^2	p-value	χ^2	p-value	χ^2	p-value	χ^2	p-value
GAT [39]	4.9123	0.0289	4.3062	0.0379	4.4690	0.0393	5.4038	0.0268
FinGAT [40]	5.8667	0.0214	3.9858	0.0428	3.8873	0.0456	5.3990	0.0301
AD-GAT [41]	5.0141	0.0251	4.8421	0.0277	5.8967	0.0283	6.3581	0.0217
SPLSTM [24]	7.9150	0.0050	6.0890	0.0136	7.7795	0.0052	8.2434	0.0034
TLSTM [9]	9.0638	0.0026	7.9399	0.0048	9.7556	0.0017	7.4654	0.0069
CLSR [38]	3.7817	0.0463	3.7630	0.0413	4.2043	0.0403	3.9203	0.0377
OURS	8.0554	0.0045	9.6761	0.0018	9.4595	0.0026	7.8881	0.0049

dataset, our method further demonstrates superiority, with its ACC of 0.5463, Precision of 0.5609, F1-score of 0.6065, and Recall reaching 0.6703, especially in the Recall metrics, which are significantly better than the other methods. This indicates that our method not only has high classification accuracy, but also can effectively identify positive examples, showing good adaptability and robustness when dealing with financial time series forecasting tasks. This result verifies the effectiveness and superiority of the method in this paper in the field of financial forecasting, and provides strong support for subsequent research and practical applications.

5.4. Statistical test

To verify the validity of the models, a Chi-square test is conducted on four benchmark datasets (CSI 500, Nasdaq, FTSE 100, and CMIN-US).

From Table 5, the chi-square test results indicate that all models demonstrated statistically significant predictive capabilities across the datasets ($p < 0.05$), rejecting the null hypothesis of randomness. Our method consistently achieved the strongest statistical significance, with p-values uniformly below 0.005 (range: 0.0018–0.0049), reflecting robust deviations from random predictions. While some baseline models such as TLSTM and SPLSTM exhibited high significance on specific datasets, the narrow p-value range of our method highlights its cross-dataset stability. This statistical consistency further validates the reliability of our method in diverse market environments.

5.5. Ablation study

Throughout our framework, the key parts of our model are the AHCE, the HHSA and the HDE, we need to figure out the particular effect of each module. Thus, in this part, we test our method by setting each part of our framework differently.

Ablation on the HHS: To assess the effectiveness of the HHS module, we conduct ablation experiments on the number of modules (i.e. TRPCA layers). It can be found from Table 6 that as the number of TRPCA layers increases, indicators such as prediction accuracy also increase, which shows that HHS plays a key role in the separation of homogeneous pattern and heterogeneous pattern in prediction. However, the number of TRPCA layers is constrained by the amount of data. In the experiment, when we set the number of TRPCA layers to 4, the results are not as good as when the number of TRPCA layers is 3.

Ablation on the AHCE: As shown in Table 7, we can see that when there is only the AHCE module, the prediction accuracy and other indicators are average. Although the AHCE part has not yet taken into account the heterogeneous information part, when AHCE and other modules are combined, the prediction accuracy and other indicators are significantly improved, which can illustrate the effectiveness of AHCE to a certain extent, and also illustrate the effectiveness of separating homogeneous and heterogeneous pattern after the AHCE module.

Ablation on the HHSA: In this section, we mainly ablate the homogeneous pattern of HHSA to facilitate comparison with the heterogeneous pattern of the HDE module. Based on the results shown in Table 7, we can observe that the HHSA module has the highest improvement in prediction accuracy, which shows the importance of homogeneous pattern in prediction accuracy, and further shows that the relationship between companies in a short period of time is homogeneous.

Ablation on the HDE: According to the results in Table 7, we can find that when there is no HDE module, each prediction index has a certain degree of decline, which proves that the heterogeneous pattern mined by HDE plays an indispensable role in the entire prediction process. At the same time, the HDE module also fills the gap of the AHCE module's lack of attention to heterogeneous pattern.

5.6. Investments simulation

To evaluate the effectiveness of our model, we conducted a stock investment simulation using CSI500 data from October to December 2015. This approach allowed us to assess the model's performance over a specific period while minimizing the risk of overfitting or data snooping bias. We compare our method with five state-of-the-art methods discussed in Section 5.3 in real investment scenarios. The initial capital is set at CNY 50,000, and we compare the cumulative daily returns based on continuous investment. In the simulation, we disregard transaction fees and perform daily buy/sell operations. Investment options are selected based on the predicted probabilities of daily price movements ranked by our framework. When sufficient funds are available, we prioritize purchasing the top five stocks with the highest predicted probabilities. Fig. 7 illustrates the cumulative returns over time.

Fig. 7 presents the investment simulations of our approach and compares it with several existing methods, including GAT, FinGAT, AD-GAT, SPLSTM, TLSTM, and CLSR. As demonstrated in the accompanying figure, our approach consistently exhibits superior performance in comparison to alternative methods over the 60 day investment period. Specifically, the proposed approach yielded a return rate of 12.84%, with the maximum return being RMB 56,418, which is considerably higher than the returns of other methods (GAT: 5.57%, FinGAT: 5.09%, AD-GAT: 4.62%, SPLSTM: 8.35%, TLSTM: 7.68%, and CLSR: 4.86%). Statistical analysis indicates that the difference in returns between our approach and the other methods is statistically significant (p-values < 0.05).

During the initial phase of the investment period, all methods show some degree of volatility. However, our approach demonstrates a more stable and steady upward trend compared to the other methods. This suggests that our approach is better at capturing market trends and making effective investment decisions in the early stages of the investment period. In the mid-phase, the performance of the different methods begins to diverge more significantly. While some methods experience fluctuations and even declines in portfolio value, our approach continues to show a consistent upward trajectory. This further highlights the robustness and effectiveness of our approach in navigating market dynamics and maximizing returns. As the investment period progresses towards its conclusion, our approach continues to outperform the other methods, reaching the highest portfolio value of CNY 56,418. This demonstrates the superior long-term performance and profitability of our approach.

Table 6
Prediction performance under different numbers of TRPCA layers. The **top** value is emphasized using **bold text**.

TRPCA layers	CSI500 data set				Nasdaq data set			
	Acc	Precision	F1-score	Recall	Acc	Precision	F1-score	Recall
1	0.5607	0.5327	0.5392	0.5527	0.5246	0.5345	0.5354	0.5492
2	0.5931	0.5412	0.5725	0.6145	0.5374	0.5408	0.5483	0.5566
3	**0.6040**	**0.5454**	**0.5876**	**0.6231**	**0.5481**	**0.5469**	**0.5993**	**0.6690**
4	0.5823	0.5371	0.5653	0.5992	0.5410	0.5423	0.5804	0.6357

Table 7
Quantitative results of different versions of our proposed model on the CSI500 and Nasdaq data sets. The **top** and second highest values are respectively emphasized using **bold text** and underlining.

Module			CSI500 data set				Nasdaq data set			
AHCE	HHSA	HDE	Acc	Precision	F1-score	Recall	Acc	Precision	F1-score	Recall
✓			0.5505	0.5167	0.5314	0.5692	0.5235	0.5347	0.5322	0.5431
	✓		0.5708	0.5335	0.5729	0.6202	0.5376	0.5413	0.5574	0.5835
		✓	0.5640	0.5187	0.5612	0.6107	0.5322	0.5392	0.5468	0.5647
✓	✓		0.5721	0.5357	0.5711	0.6144	0.5417	0.5484	0.5482	0.5597
✓		✓	0.5539	0.5389	0.5615	0.5830	0.5396	0.5434	0.5370	0.5898
	✓	✓	0.5850	0.5361	0.5742	**0.6271**	0.5419	**0.5531**	0.5865	0.6253
✓	✓	✓	**0.6040**	**0.5454**	**0.5876**	0.6231	**0.5481**	0.5469	**0.5993**	**0.6690**

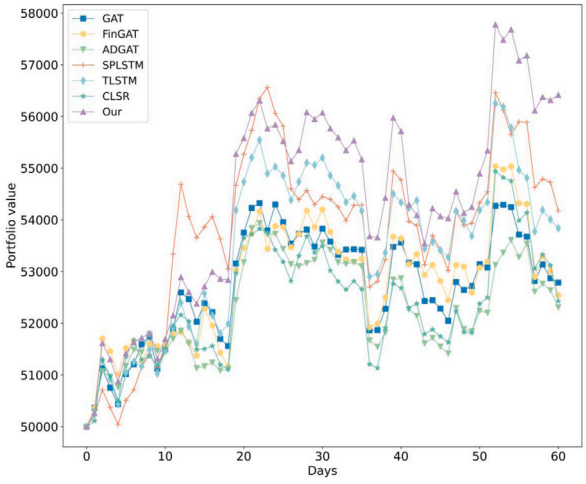


Fig. 7. Short-term investment simulations.

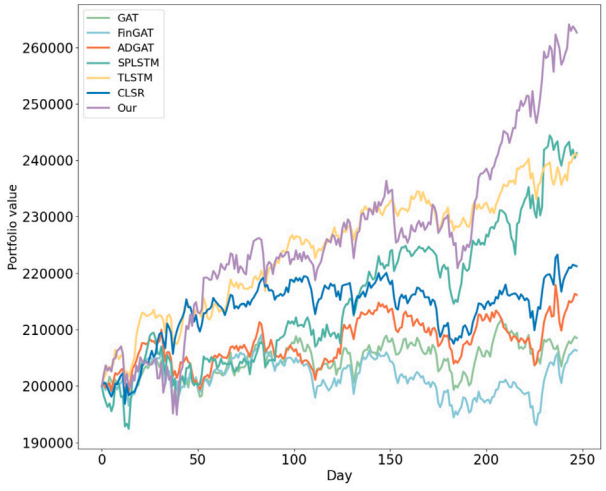


Fig. 8. Long-term investment simulations.

In order to verify the effectiveness of the strategy in different market cycles, a long-term backtest was conducted based on the CMIN-US dataset. The initial capital is set at USD 200,000, and we compare the cumulative daily returns based on continuous investment. In the simulation, we disregard transaction fees and perform daily buy/sell operations. Investment options are selected based on the predicted probabilities of daily price movements ranked by our framework. When sufficient funds are available, we prioritize purchasing the top five stocks with the highest predicted probabilities. The experiment involved the selection of trading day data from January 1 to December 31, 2021 for the purpose of investment backtesting. This period was a special year under the unprecedented circumstances of the COVID-19 pandemic, and the data was unique and representative. The uncertainty engendered by the pandemic has facilitated a more comprehensive evaluation of the performance and adaptability of strategies in such contexts, ensuring that experimental data encompasses a broader spectrum of risk changes than most previous studies. The backtest results are shown in Fig. 8.

As demonstrated in Fig. 8, the efficacy of the our strategy is evident in its superior performance in the evaluation of stock investment strategies, outperforming alternative strategies. The portfolio value shows a consistent and stable growth trajectory over the 250 day investment cycle. During the initial 150 day period of the investment cycle, the portfolio value of each strategy exhibits a relatively similar growth trend. However, commencing on the 150th day, our strategy initiates a gradual widening of the gap with competing strategies. This acceleration peaks at the 200th day, ultimately resulting in a portfolio value of approximately 245,000 on the 250th day. In contrast, although the SPLSTM strategy (dark green curve) performs second best, its portfolio value only reaches approximately 225,000 on the 250th day, which is somewhat different from our strategy. Concurrently, the portfolio values of alternative strategies, such as GAT, FinGAT, and ADGAT, are essentially below 220,000 on the 250th day, and the aggregate performance is comparatively inadequate.

In terms of volatility, our strategy also shows a smaller fluctuation range, indicating that it has a stronger ability to resist risks. The volatility of other strategies is relatively large, especially in certain periods of volatile market conditions, when the value of their portfolios has fallen significantly, which may indicate that these strategies have certain deficiencies in risk control and response to market changes.

With regard to investment returns, our strategy has achieved a cumulative return of about 27.24% during the investment cycle, while the TLSTM strategy has a cumulative return of about 18.76%, the SPLSTM strategy has a cumulative return of about 18.63%, and the cumulative

returns of other strategies are below 15%, showing our significant advantage in investment returns. Statistical analysis indicates that the difference in returns between our approach and the other methods is statistically significant (p-values < 0.05). At the same time, our strategy has maintained a high rate of return on investment for most of the time period, and can more effectively capture investment opportunities in the stage of positive market trends, thereby achieving a rapid increase in the value of the portfolio.

6. Discussion

6.1. Theoretical and practical implications

The theoretical foundation can be traced back to the exact separation of the low-rank and sparse components under certain conditions in [19]. Wang et al. [9] have empirically explored that this low-rank and sparse modeling, i.e., the TRPCA model, is practical for securities market data. On this basis, we further unroll their iterative optimization algorithm for the TRPCA model into a deep neural network, with each module being well in accordance with the algorithm's updating. Thus, we obtain a HHS, with both interpretability and a higher model capacity for the separation of homogeneous and heterogeneous information. This indeed expands the application scope of the mathematical theory in [20] to the field of financial data.

Then, we fully explore the homogeneous and heterogeneous information by designing a novel self-attention mechanism. This mechanism can make full use of both homogeneous and heterogeneous data in a reasonable and effective way, yielding better prediction results. Therefore, our work not only enriches the application of the theory in [19], but also further provides an elegant way to utilize those homogeneous and heterogeneous components for the subsequent prediction task.

6.2. Limitations

In this study, the explicit mapping of the optimization process is achieved through the algorithmic unrolling network technique, and the self-attention mechanism is customized based on the interpretable output of the unrolling network. This design significantly enhances the interpretability of our method. However, to fully capture the characteristics of the financial market data, we introduce a self-attention-based LSTM for prediction, which remains a black box, limiting the whole interpretability of our work. Nonetheless, as financial markets' fluctuations in stock price are influenced by a complex interplay of historical trends, event sequences, and market sentiment, exhibiting nonlinear and time-varying characteristics, introducing the LSTM or other deep neural network structures to modeling the complex nonlinear patterns is inevitable.

6.3. Future work

While our current framework achieves partial interpretability through the algorithm unrolling, the predictive module remains a black-box that requires deeper structural transparency. Recent advances in white-box architectures show promising directions. For instance, the Transformer variant built upon sparse coding principles attained 85.1% image classification accuracy while maintaining architectural interpretability [41], particularly through its rate reduction objective that enables mathematically analyzable feature transformations. Building on these insights, we plan to derive a white-box attention mechanism by unrolling optimization procedures for sparse coding, which may reveal the mathematical principles governing cross-modal information fusion in stock prediction.

7. Conclusion

This paper proposes an interpretable model-guided stock prediction framework that integrates homogeneous–heterogeneous processing using unrolled TRPCA and a customized self-attention mechanism. The key innovation of this framework is that it can effectively combine unrolled principal component analysis and customized self-attention mechanism. Specifically, the unrolled TRPCA effectively separates low-rank homogeneous components and sparse heterogeneous components, while the customized self-attention mechanism models homogeneous information and heterogeneous information. This architectural innovation achieves an organic balance between mathematical interpretability and deep learning representation capabilities. The unrolled TRPCA module provides an interpretable homogeneous and heterogeneous decoupling process, while the self-attention mechanism better captures the global dependencies of homogeneous information and the local correlation patterns of heterogeneous information. The synergy of the two enables the model to achieve excellent prediction performance while maintaining a certain degree of interpretability. The proposed method significantly improves prediction accuracy on the CSI 500, Nasdaq and FTSE 100 datasets, outperforming state-of-the-art methods while maintaining partial interpretability of the results.

CRediT authorship contribution statement

Yi Zhou: Writing – original draft, Software, Methodology, Data curation, Conceptualization. **Tai-Xiang Jiang:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization. **Jun Wang:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization. **Jinghua Tan:** Writing – original draft, Methodology, Funding acquisition, Data curation, Conceptualization.

Declaration of competing interest

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

Acknowledgments

The authors would like to thank the editor and reviewers for giving them many comments and suggestions, which are of great value for improving the quality of this manuscript. This work was supported in part by Sichuan Science and Technology Program under Grant 2024ZYD0147, in part by Natural Science Foundation of Xinjiang Uygur Autonomous Region under Grant 2024D01A18, in part by National Natural Science Foundation of China (NSFC) under Grant 72301188, and in part by China Postdoctoral Science Foundation under Grants 2023M732503 and 2025T181014.

Appendix A. Mathematical foundations

Definition 1 (*T-product [35]*)**.**: The tensor–tensor product $\mathcal{C} = \mathcal{A} * \mathcal{B}$ of tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ results in a tensor of size $n_1 \times n_4 \times n_3$. The (i,j)th tube $\mathbf{c}_{ij:}$ is determined by

$$\mathbf{c}_{ij:} = C(i,j,:) = \sum_{g=1}^{n_2} \mathcal{A}(i,g,:) * \mathcal{B}(g,j,:) \tag{A.1}$$

where $*$ represents the circular convolution between two tubes of identical size. An effective method for computing the t-prod of two tensors involves the following steps: (i) applying a Fast Fourier transform (FFT) along tubes, (ii) conducting matrix multiplications of each pair of frontal slices of the tensors in the transform domain, and (iii) applying an inverse FFT along tubes of the resulting tensor.

Definition 2 (*Conjugate Transpose [35]*). The conjugate transpose of a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, denoted as $\mathcal{T}^H$, is defined as follows:

(i) $(\mathcal{T}^H)^{(1)} = (\mathcal{T}^{(1)})^H$

(ii) $(\mathcal{T}^H)^{(i)} = (\mathcal{T}^{(n_3+2-i)})^H$, where $i = 2, \ldots, n_3$.

Theorem 1 (*t-SVD [35]*).: *For a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the t-SVD of $\mathcal{T}$ is expressed as*

$$\mathcal{T} = \mathcal{U} * \mathcal{S} * \mathcal{V}^H \tag{A.2}$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors which satisfy $\mathcal{U} * \mathcal{U}^H = \mathcal{U}^H * \mathcal{U} = \mathcal{V} * \mathcal{V}^H = \mathcal{V}^H * \mathcal{V} = \mathcal{I}$, and $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor which each frontal slice $\mathcal{S}^{(i)}$ is a diagonal matrix.

Definition 3 (*Tensor tubal rank [42]*).: Given the t-SVD: $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^H$, where $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the tubal rank $rank_t(\mathcal{A})$ the count of nonzero singular tubes in $\mathcal{S}$.

Definition 4 (*Tensor Tubal Nuclear Norm (TNN) [42]*).: Denoted as $\|\mathcal{X}\|_{\text{TNN}}$, the tensor nuclear norm of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined as

$$\|\mathcal{X}\|_{\text{TNN}} = \sum_{i=1}^{n_3} \|\tilde{\mathcal{X}}^{(i)}\|_* \tag{A.3}$$

where $\tilde{\mathcal{X}}$ refers to the Fourier transformed tensor along the third mode.

Minimizing TNN effectively promotes tensor low-rank properties [43].

Appendix B. Details of Algorithm 1

The iterative workflow of Algorithm 1 is detailed as follows.

1. Algorithm parameter description
   - $\mathcal{L}^{(0)}$: Initial for the low-rank component (default: $\mathcal{X}$).
   - $\mathcal{T}^{(0)}$: Initial for the sparse component (default: zero tensor).
   - $\mathcal{M}^{(0)}$: Lagrangian multiplier enforcing the constraint $\mathcal{X} = \mathcal{L} + \mathcal{T}$ (default: zero tensor).
   - $\beta > 0$: Penalty parameter controlling the trade-off between constraint satisfaction and objective minimization.
   - $\lambda \geq 0$: Regularization parameter for the sparse component, embedded in the shrinkage operator.
   - Convergence thresholds $\eta_{\mathcal{L}}, \eta_{\mathcal{T}} > 1 \times 10^{-6}$: Relative error tolerances for terminating iterations.
2. Low-Rank Component $\mathcal{L}$ Update (Lines 2–3): The low-rank component $\mathcal{L}$ is recovered by minimizing the TNN. The process is as follows:
   - Let $\mathcal{L} = \mathcal{X} - \mathcal{T}^{(k)} + \frac{\mathcal{M}^{(k)}}{\beta}$. By performing t-SVD on $\mathcal{L}$, we obtain:

     $$\mathcal{L} = \mathcal{U} * \mathcal{S} * \mathcal{V}^H \quad , \tag{B.1}$$

     where $\mathcal{U}, \mathcal{V}$ are orthogonal tensors and $\mathcal{S}$ is the f-diagonal singular value tensor. Based on this decomposition, the closed-form update for $\mathcal{L}$ is given by:

     $$\mathcal{L}^{(k+1)} = \mathcal{U} * \text{Shrink}_{1/\beta}(\mathcal{S}) * \mathcal{V}^H, \tag{B.2}$$

     with the shrinkage operator defined as:

     $$\text{Shrink}_{\tau}(s) = \text{sign}(s) \cdot \max(|s| - \tau, 0). \tag{B.3}$$

   These steps ensure that $\mathcal{L}$ maintains its low-rank property, thereby effectively separating the low-rank component.
3. Sparse Component $\mathcal{T}$ Update (Line 4):
   - Compute the update using entry-wise shrinkage:

     $$\mathcal{T}^{(k+1)} = \text{Shrink}_{\frac{\lambda}{\beta}}\left(\mathcal{X} - \mathcal{L}^{(k+1)} + \frac{\mathcal{M}^{(k)}}{\beta}\right), \tag{B.4}$$

     where the shrinkage threshold $\frac{\lambda}{\beta}$ controls sparsity.

   This step eliminates small entries in the residual to model sparse outliers or noise.
4. Lagrangian Multiplier Update (Line 5):
   - Adjusts the multiplier to penalize deviations from the constraint $\mathcal{X} = \mathcal{L} + \mathcal{T}$, ensuring feasibility as β increases.
5. Convergence Check (Lines 6–7):
   - Calculate relative residuals for $\mathcal{L}$ and $\mathcal{T}$ updates. The residuals measure the relative change between consecutive iterations. A small term (10^{-10}) is added to avoid division by zero. The loop stops if the maximum of these residuals falls below 1×10^{-6}.

Data availability

Data will be made available on request.

References

[1] M. Usmani, S.H. Adil, K. Raza, S.S.A. Ali, Stock market prediction using machine learning techniques, in: Proceedings of the International Conference on Computer and Information Sciences, 2016, pp. 322–327.

[2] T. Yin, C. Liu, F. Ding, Z. Feng, B. Yuan, N. Zhang, Graph-based stock correlation and prediction for high-frequency trading systems, Pattern Recognit. 122 (2022) 108209.

[3] S. Liao, L. Xie, Y. Du, S. Chen, H. Wan, H. Xu, Stock trend prediction based on dynamic hypergraph spatio-temporal network, Appl. Soft Comput. 154 (2024) 111329.

[4] K.-j. Kim, Financial time series forecasting using support vector machines, Neurocomputing 55 (1–2) (2003) 307–319.

[5] R.P. Schumaker, H. Chen, A quantitative stock prediction system based on financial news, Inf. Process. Manage. 45 (5) (2009) 571–583.

[6] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, T.-S. Chua, Temporal relational ranking for stock prediction, ACM Trans. Inf. Syst. 37 (2) (2019) 1–30.

[7] D. Cheng, F. Yang, S. Xiang, J. Liu, Financial time series forecasting with multi-modality graph neural network, Pattern Recognit. 121 (2022) 108218.

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

[9] J. Wang, Y. Hu, T.-X. Jiang, J. Tan, Q. Li, Essential tensor learning for multimodal information-driven stock movement prediction, Knowl.-Based Syst. 262 (2023) 110262.

[10] Y. Zhao, G. Yang, Deep learning-based integrated framework for stock price movement prediction, Appl. Soft Comput. 133 (2023) 109921.

[11] X. Zhang, Y. Zhang, S. Wang, Y. Yao, B. Fang, S.Y. Philip, Improving stock market prediction via heterogeneous information fusion, Knowl.-Based Syst. 143 (2018) 236–247.

[12] H. Ince, T.B. Trafalis, Short term forecasting with support vector machines and application to stock price prediction, Int. J. Gen. Syst. 37 (6) (2008) 677–687.

[13] F. Ming, F. Wong, Z. Liu, M. Chiang, Stock market prediction from WSJ: Text mining via sparse matrix factorization, in: Proceedings of the IEEE International Conference on Data Mining, 2014, pp. 430–439.

[14] R. Singh, S. Srivastava, Stock prediction using deep learning, Multimedia Tools Appl. 76 (2017) 18569–18584.

[15] W. Kong, F. Zhang, W. Qin, J. Wang, Low-tubal-rank tensor recovery with multilayer subspace prior learning, Pattern Recognit. 140 (C) (2023) 109545.

[16] M.K. Ng, X. Zhang, X.-L. Zhao, Patched-tube unitary transform for robust tensor completion, Pattern Recognit. 100 (2020) 107181.

[17] Q. Li, L. Jiang, P. Li, H. Chen, Tensor-based learning for predicting stock movements, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2015, pp. 1784–1790.

[18] Q. Li, Y. Chen, L.L. Jiang, P. Li, H. Chen, A tensor-based information framework for predicting the stock market, ACM Trans. Inf. Syst. 34 (2) (2016) 1–30.

[19] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, S. Yan, Tensor robust principal component analysis with a new tensor nuclear norm, IEEE Trans. Pattern Anal. Mach. Intell. 42 (4) (2019) 925–938.

[20] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: From error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[21] A.A. Ariyo, A.O. Adewumi, C.K. Ayo, Stock price prediction using the ARIMA model, in: Proceedings of the International Conference on Computer Modelling and Simulation, 2014, pp. 106–112.
[22] T. Bollerslev, Glossary to ARCH (GARCH), CREATES Res. Pap. 49 (2008).
[23] J. Jiang, L. Wu, H. Zhao, H. Zhu, W. Zhang, Forecasting movements of stock time series based on hidden state guided deep learning approach, Inf. Process. Manage. 60 (3) (2023) 103328.
[24] 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.
[25] X. Li, P. Wu, W. Wang, Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong, Inf. Process. Manage. 57 (5) (2020) 102212.
[26] J. Moon, M. Jeon, S. Jeong, K.-Y. Oh, RoMP-transformer: Rotational bounding box with multi-level feature pyramid transformer for object detection, Pattern Recognit. 147 (2024) 110067.
[27] K. Gregor, Y. LeCun, Learning fast approximations of sparse coding, in: Proceedings of the International Conference on Machine Learning, 2010, pp. 399–406.
[28] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, T.S. Huang, D3: Deep dual-domain based fast restoration of JPEG-compressed images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2764–2772.
[29] Y. Yang, J. Sun, H. Li, Z. Xu, Deep ADMM-Net for compressive sensing MRI, in: Proceedings of the International Conference on Neural Information Processing Systems, 2016, pp. 10–18.
[30] J. Zhang, B. Ghanem, ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1828–1837.
[31] Z. Hu, W. Liu, J. Bian, X. Liu, T.-Y. Liu, Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction, in: Proceedings of the ACM International Conference on Web Search and Data Mining, 2018, pp. 261–269.
[32] X.-H. Dang, S.Y. Shah, P. Zerfos, "The Squawk Bot": Joint learning of time series and text data modalities for automated financial information filtering, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2021, pp. 4597–4603.
[33] S. Li, W. Liao, Y. Chen, R. Yan, PEN: Prediction-explanation network to forecast stock price movement with better explainability, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023, pp. 5187–5194.
[34] M.E. Kilmer, C.D. Martin, Factorization strategies for third-order tensors, Linear Algebra Appl. 435 (3) (2011) 641–658.
[35] M.E. Kilmer, K. Braman, N. Hao, R.C. Hoover, Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging, SIAM J. Matrix Anal. Appl. 34 (1) (2013) 148–172.
[36] S. Boyd, N. Parikh, E. Chu, Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers, Now Publishers Inc, 2011.
[37] W. Feng, X. Ma, X. Li, C. Zhang, A representation learning framework for stock movement prediction, Appl. Soft Comput. 144 (2023) 110409.
[38] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, et al., Graph attention networks, in: Proceedings of the International Conference on Learning Representations, 2018, pp. 2920–2931.
[39] Y.-L. Hsu, Y.-C. Tsai, C.-T. Li, FinGAT: Financial graph attention networks for recommending top-K profitable stocks, IEEE Trans. Knowl. Data Eng. 35 (1) (2021) 469–481.
[40] R. Cheng, Q. Li, Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 55–62.
[41] J. Yang, X. Li, D. Pai, Y. Zhou, Y. Ma, Y. Yu, C. Xie, Scaling white-box transformers for vision, Adv. Neural Inf. Process. Syst. 37 (2024) 36995–37019.
[42] Z. Zhang, G. Ely, S. Aeron, N. Hao, M. Kilmer, Novel methods for multilinear data completion and de-noising based on tensor-SVD, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3842–3849.
[43] Y.-S. Luo, X.-L. Zhao, T.-X. Jiang, Y. Chang, M.K. Ng, C. Li, Self-supervised nonlinear transform-based tensor nuclear norm for multi-dimensional image recovery, IEEE Trans. Image Process. 31 (2022) 3793–3808.