RESEARCH ARTICLE

Multi-perspective Learning Based on Transformer for Stock Price Trend

Xiliang Li[1] · Shuoru Chen[2] · Xiaoyan Qiao[1,3,4] · Mingli Zhang[2] · Caiming Zhang[2] · Feng Zhao[2,3,4]

Received: 21 September 2024 / Accepted: 7 February 2025

Abstract
Stock constitutes a crucial element of the financial market, and accurately forecasting stock trends remains a significant and unresolved issue. Nonetheless, the stock's considerable complexity renders accurate prediction of stock trends more challenging. This paper proposes a novel multi-perspective approach that converts the time series prediction challenge into an image classification problem, referred to as the Multi-perspective Denoise Transformer (MPDTransformer). We initially multi-factor features into two-dimensional images employing a multi-perspective approach to more comprehensively explain the actual market conditions and enhance the model's practicality and adaptability; secondly, we utilize a Convolutional Autoencoder (CAE) to extract features, which effectively eliminates noise and enhances data purity; finally, to comprehensively capture the temporal relationships within the data and gain a deeper understanding of the overall time series, we employ a Transformer for prediction. Experimental results demonstrate that our method outperforms other prevalent stock trend prediction techniques.

Keywords Technical indicators · Multi-factor model · Transformer · Convolutional autoencoder (CAE) · Stock trend prediction

1 Introduction

Stock prediction, which anticipates future market trends and price fluctuations based on stock market dynamics, is essential for investors to grasp market trends, optimize portfolios, and mitigate risks. In theory, deep learning-based stock prediction techniques enhance the investigation of financial theories and enrich the academic foundation. They more precisely capture the complex nature of time-series data, thereby improving prediction accuracy and reliability. These behaviors are essential for national economic stability, corporate governance, and individual investment decisions. A stable stock market signifies a nation's political and economic conditions, fosters economic growth and job creation, and ensures financial stability [1]. For corporations, stock market stability enhances fundraising for growth and innovation. Accurate predictions empower individual investors to make informed investment decisions, preserve assets, and optimize returns. Stock price prediction is affected by multiple factors, including market sentiment, company performance, investor behavior, and external uncertainties such as the macroeconomic environment, national policies, natural disasters, and social events. The interaction of complexity, volatility, and uncertainty complicates the accurate prediction of stock market trends, posing a considerable challenge. Researchers have concentrated extensively on predicting stock price trends and utilized

 https://doi.org/10.1007/s44196-025-00768-w

Published online: 27 February 2025

 https://doi.org/10.1007/s44196-025-00768-w

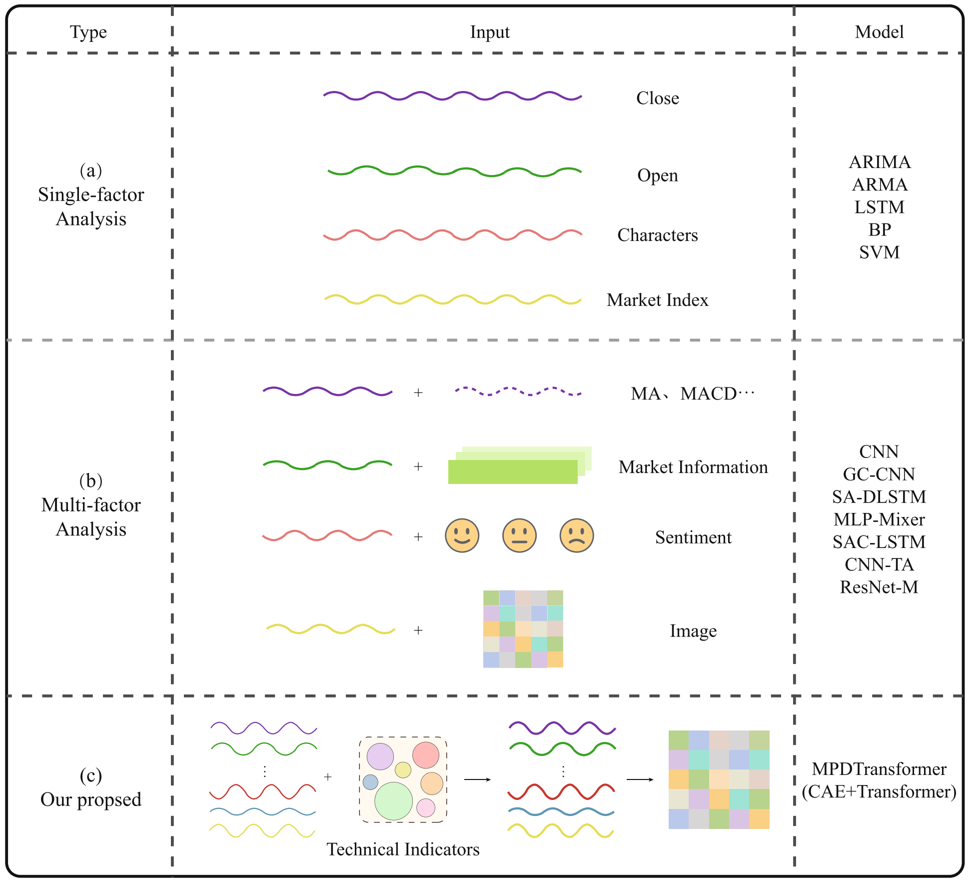


Fig. 1 Different model architectures **a** Some methods for single-factor analysis **b** Some methods for multi-factor analysis **c** Our proposed method

diverse methodologies to analyze trends from various perspectives [2–6]. Research methodologies can be broadly categorized into two primary types: single-factor analysis and multi-factor analysis.

Single-factor analysis examines the influence of individual variables on fluctuations in stock prices, including the closing price, opening price, market indices, and similar factors. For instance, as illustrated in Fig. 1a, the studies [7–9] employed traditional models including Autoregressive Integrated Moving Average (ARIMA), Autoregressive Moving Average (ARMA), Long Short-Term Memory (LSTM), and BP neural network to forecast the close price time series data of the stock market. Meanwhile, numerous studies focus on investigating the trends of stock opening prices. Khandelwal [10] and Kim [11] employed neural networks, ARIMA, and Support Vector Machine (SVM) methods, respectively, to predict the open prices of time series data, thereby significantly improving the predictive accuracy of their models. In addition to the basic forecasts of closing and opening prices, certain studies have performed predictive analyses on particular stock indices. Hafezi et al. [12] employed cross-correlation feature methods and support vector machines to select 13 features for predicting the stock market. Xiong [13] and Fischer [14] utilized LSTM networks to forecast the volatility of the S&P 500 index. Liu et al. [15] proposed a framework based on LSTM for forecasting stock market trends, validating the efficacy of this approach using actual data from the S&P 500 Index, DJIA, and China Minsheng Bank. Although these studies have enhanced prediction accuracy to some degree, the financial market is influenced by numerous factors, making the single-factor model insufficient. Therefore, a more adaptable and flexible model is urgently needed to efficiently tackle the complexities of financial markets.

In reaction to the limitations of single-factor models, researchers have shifted from one-dimensional to more comprehensive two-dimensional multi-factor models. This evolution involves the integration of various elements, including technical indicators, fundamental data, and market sentiment, to thoroughly capture market dynamics. Numerous studies have incorporated various factors from input datasets. For instance, as illustrated in Fig. 1b, Hoseinzad utilizes target stocks as inputs to forecast future stock trends employing a Convolutional Neural Network. Wei Chen et al. [16] proposed a Graph Convolutional Convolutional Neural Network (GC-CNN) model that incorporated both market-wide and individual stock data. Yang et al. [17] integrated multi-view stock data with a Multi-Perceptron Mixer (MLP-Mixer) to forecast stock trends. Certain studies have utilized sentiment analysis (SA), Denoising Autoencoder (DAE) models, and LSTM networks in conjunction (SA-DLSTM [18]) or integrated Convolutional Neural Networks with LSTM (SAC-LSTM [19])) to analyze historical data and financial information, converting them into image data for predictive purposes. Sezer et al. [1] presented the CNN-TA algorithm trading model, which converts financial time series into images and employs CNN for forecasting. Lin et al. [20] proposed a Residual Network (ResNet-M) model for predicting stock price trends based on graphical features of stock data. Liu et al. [5] conceptualized stock prices as images and utilized Deep Learning Neural Networks (DLNN) for image modeling, leveraging price charts and fundamental stock data to predict short-term stock price trends. Gao et al. [21] employed PCA for the dimensionality reduction of technical indicators, subsequently integrating the fundamental trading data with the reduced technical indicators into a designed LSTM model to forecast the closing prices of the stock market; Carosia et al. [22] introduced a model that combines historical stock prices, financial technical indicators, and financial news to predict the Brazilian stock market; Wang et al. [23] developed a three-stage hybrid model that incorporates daily sentiment and technical indicators for stock trend predictions, utilizing deep learning LSTM and CNN as the principal predictive elements at various stages; Ma et al. [24] utilized 27 technical indicators and 5 raw price sequences as inputs to a CNN model to predict the subsequent day's stock price movements, with experimental results demonstrating that this multi-input model achieved an average prediction accuracy of about 70 percent for both stock market indices and individual stocks.

Recent studies have thoroughly documented the effectiveness of Convolutional Autoencoders (CAE) in denoising financial data. The application of CAE for hyperspectral unmixing illustrates its adaptability in managing significant noise disturbances, rendering it especially appropriate for financial time series data [25]. Likewise, the Transformer architecture has demonstrated considerable benefits in capturing complex temporal relationships in time series data. Researches, including the Modality-aware Transformer [26] for financial time series forecasting, highlight the model's proficiency in accurately capturing both long-term and short-term dependencies, thus enhancing prediction accuracy. The Transformer's global modeling capability and adaptability to dynamic data fluctuations make it an outstanding instrument for stock trend forecasting.

Recent studies have investigated advanced optimization techniques in portfolio selection. Leung et al. [27] proposed a collaborative neurodynamic optimization technique for cardinality-constrained portfolio selection, framing the issue as a mixed-integer optimization problem and solving it through multiple recurrent neural networks adjusted by a particle swarm optimization rule. This approach effectively resolves the dilemma of balancing diversification and concentration in investment portfolios. Leung and Wang developed a minimax and biobjective portfolio selection model utilizing collaborative neurodynamic optimization, which employs multiple neural networks to define the efficient frontier through particle swarm optimization-based weight optimization. These studies emphasize the effectiveness of neurodynamic optimization for solving complex financial challenges and offer significant insights for our research on stock prediction utilizing advanced optimization methodologies.

However, despite the fact that these methods forecast and provide appropriate experimental outcomes using inputs from multiple factors and perspectives, there are some significant limitations: (1) The use of fewer multi-factor parameters reduces comprehensiveness and representativeness, potentially influencing analysis results. (2) The non-stationary and uncertainty inherent in stock data add a significant amount of noise to our input data, and insufficient denoising processing may result in deviations in prediction results.(3) When using Convolutional Neural Networks (CNN) for image processing, the global relationship between previous and subsequent points is not fully considered, which may impede the analysis of time series data.

To address the aforementioned issues, we propose a hybrid multi-factor model, MPDTransformer, as illustrated in Fig. 1(c), that can efficiently denoise stock data and allow for more precise and comprehensive analysis. This method converts common stock prediction issues into image classification tasks. Initially, we select a set of relevant and exhaustive multi-factor indicators to generate a two-dimensional dataset that closely resembles real-

 https://doi.org/10.1007/s44196-025-00768-w

world market conditions. This option increases the model's practicality and versatility. Subsequently, we use the Convolutional Autoencoder (CAE) to reduce noise in the dataset, improving its integrity. This step is critical because CAE effectively eliminates irrelevant information and noise that are common in financial data. This improves the quality of the input data, making it better suited for accurate prediction. Finally, by using Transformer for data classification, we enable a more comprehensive interpretation of data relationships. The Transformer architecture is particularly useful for predicting stock trends because it can capture long-term dependencies and global patterns in time series data. This is necessary for understanding complex financial dynamics and making accurate predictions. This global perspective improves the model's understanding of the overall time series structure, leading to more reliable and accurate predictions of future trends. To summarize, the integration of CAE and Transformer in MPDTransformer is theoretically sound. CAE's denoising capabilities, combined with Transformer's powerful trend prediction mechanisms, provide a solid foundation for handling complex financial data. This fusion improves the model's accuracy while also ensuring its applicability in real-world scenarios.

Overall, our MPDTransformer model produces the following results: (1) Unlike traditional one-dimensional datasets, our dataset is two-dimensional, containing both basic original trading data and more comprehensive and persuasive technical indicators. By taking into account all of this multidimensional information, we can better capture various aspects of stock market features. (2) Using CAE, we effectively remove noise from the data, increasing its purity and the model's ability to extract key features. This improves its robustness and accuracy. Furthermore, to the best of our knowledge, research on effectively learning meaningful features from financial time series data using CAE is limited, so our approach of using CAE to extract the features we require is highly significant. (3) The Transformer architecture examines the relationships between data points in stock time series from a broader perspective. Unlike traditional methods, which frequently rely on local patterns and short-term dependencies, the Transformer's attention mechanism enables the model to dynamically weigh the importance of various time steps, resulting in a better understanding of the overall time series structure, including long-term dependencies and global trends. This capability is especially useful in financial data, where complex interactions and nonlinear relationships are common. As a result, our model can predict future trends with greater accuracy and reliability.

The remainder of this article is organized as follows: Sect. 1 describes the research motivation. Section 2 elaborates on the proposed model and methodology. Section 3 discusses the experimental design and outcomes. Section 4 presents a market trading simulation. Finally, Sect. 5 is the conclusion section.

2 Method

This section describes the key steps in using MPDTransformer to predict stock trends. We will describe the proposed MPDTransformer's workflow, as shown in Fig. 2, which is divided into four steps.

Step 1) Feature analysis and extraction. Relevant multi-factor features are analyzed to identify those that have a significant influence on stock price trends. These features are combined with the original data to produce a dataset that includes technical indicators, ensuring the dataset's richness and validity while also laying the foundation for MPDTransformer performance.

Step 2) Image generation of multiple technical indicators. Detailed in Sect. 2.1. This step uses a fusion method to convert the multi-factor indicators into two-dimensional images. This innovative conversion not only simplifies data processing, but it also provides an intuitive input format for future deep learning models. Unlike most previous stock prediction studies, which use one-dimensional time series analysis, our approach employs a two-dimensional representation to capture complex patterns and improve prediction accuracy.

Step 3) Denoise using CAE. We use a CAE for denoising, which is a critical component of MPDTransformer, as discussed in Sect. 2.2. CAE effectively reduces noise by learning a low-dimensional representation of the input data, thereby improving data purity and the model's ability to extract critical information for accurate predictions.

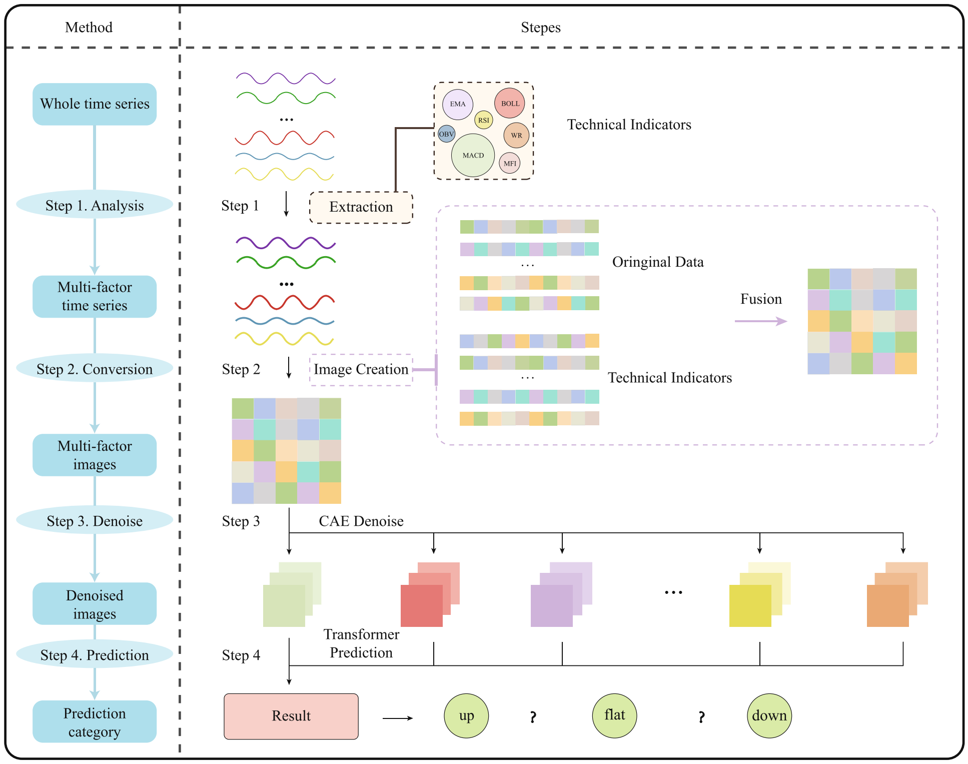


Fig. 2 Overview of MPDTransformer framework

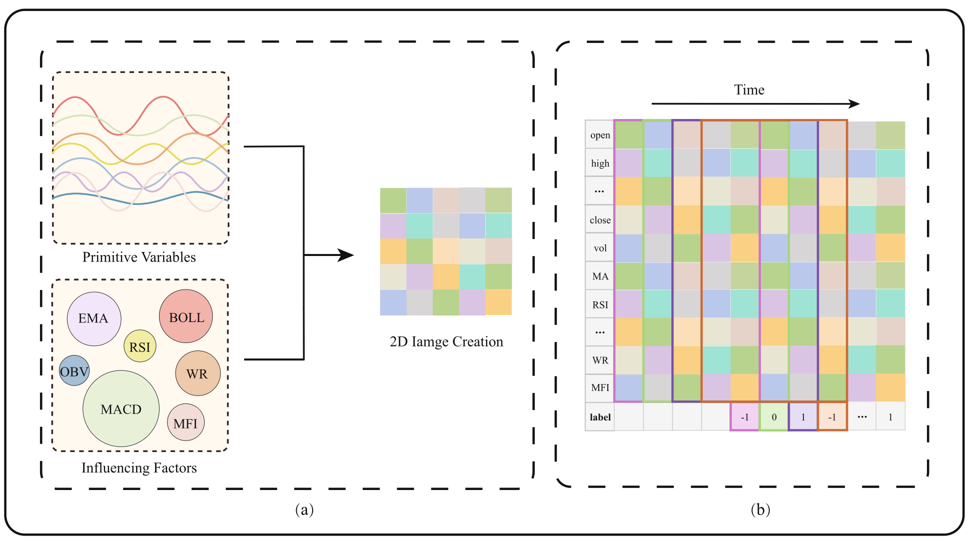


Fig. 3 Multi-factor dataset processing **a** Fusion method of multi-factor two-dimensional images **b** Label generation by sliding windows

Step 4) Prediction using Transformer. As described in Sect. 2.3, we use Transformer for trend prediction, which is another key component of MPDTransformer.The Trans-former receives features from the CAE and analyzes data correlations using a self-attention mechanism. This allows MPDTransformer to detect long-term dependencies in time series data, resulting in more accurate stock trend predictions during image classification.

2.1 Image Creation Through Merging Multi-indicators

The MPDTransformer dataset is generated by merging and transforming specific images through multi-factor features. This process involves integrating selected influential financial technical indicators $b_1, b_2, \ldots, b_m$ with daily transaction data $a_1, a_2, \ldots, a_s$ of n days. As shown in Fig. 3a, by fusing these factors, a two-dimensional

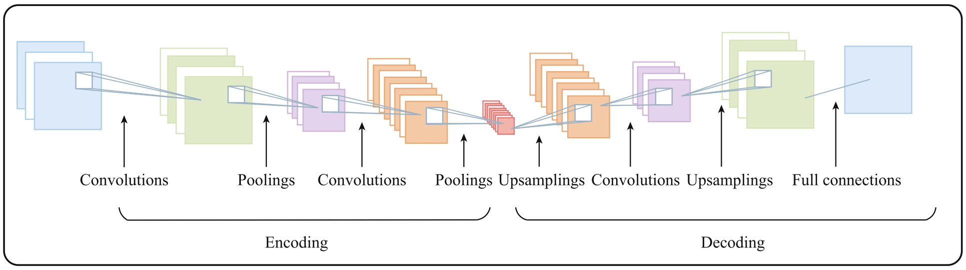


Fig. 4 The structure of convolutional autoencoder

image **A** of size $n \times (m + s)$ is produced. Financial technical indicators play a crucial role in financial market analysis, providing essential tools for traders and investors to analyze price movements and trends. Derived from historical price and volume data, these indicators are calculated through various data and statistical approaches, offering a means to measure market sentiment and strength. Categorized into trend, volatility, and momentum indicators, they assist in identifying market trends, predicting price reversals, and optimizing trade decisions, enabling the development of successful trading strategies.

Our multi-perspective fusion approach considers not only the values of individual indicators but also their interactions and relationships. This fusion strategy helps to more comprehensively interpret the real market situation, enhancing the practicality and adaptability of the model. The fused features are then converted into a two-dimensional image, where each pixel represents a specific feature value. The number of rows n in the image corresponds to the length of the time series, and the number of columns m+s corresponds to the total number of fused features. This image representation allows complex time series data to be transformed into an image classification problem, enabling analysis using deep learning models.The generated two-dimensional image A, as shown in in Fig. 3a, is a color matrix where each color block represents a specific feature value.

2.2 Denoising Based on Convolutional Autoencoder (CAE)

Convolutional Autoencoder (CAE) is an unsupervised neural network, a variant of autoencoder. It is mainly used for feature extraction and dimensionality reduction of image data, combining the feature extraction capabilities of Convolutional Neural Networks (CNN) with unsupervised learning of autoencoders to achieve representation learning of input data.

CAE consists of two parts: an encoder and a decoder, which are usually symmetrical, as shown in Fig. 4. The encoder progressively reduces the dimensionality and extracts features of the input data through a series of convolutional and pooling layers, ultimately generating a condensed representation known as the code. The task of the decoder is to decode the encoded representation into an output with the same size as the initial input. It implements this reconstruction operation through a series of deconvolution layers and upsampling operations [28]. The basic structure of CAE as shown in Fig. 4.

The primary aim of CAE is to minimize the reconstruction error, achieved by reducing the disparity between the decoded output and the initial input to maintain the reconstructed output as close as possible to the original data. In this paper, we use the Mean Squared Error (MSE) function to measure the difference between the decoder output and the original input, denoted as:

$$MSE = \frac{1}{2n} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2 \tag{1}$$

where n is the number of samples, x_i is the true value and the $\hat{x}_i$ is the predicted value.

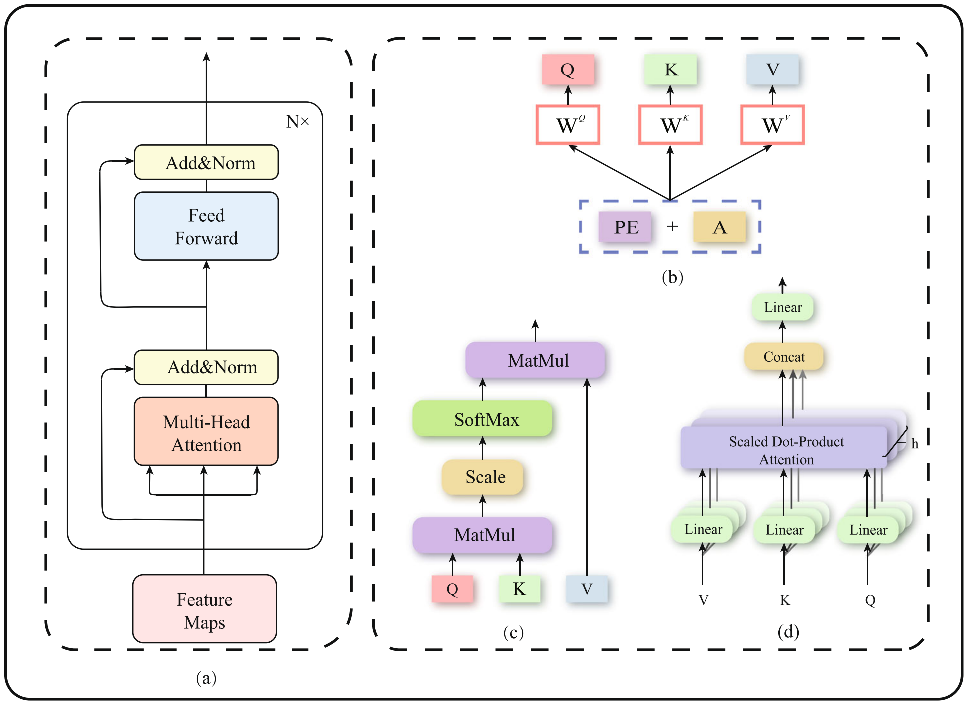


Fig. 5 Main modules of Transformer **a** The structure of Transformer encoder block **b** The calculation of embedding and positional encoding **c** The structure of dot-Product attention. **d** The structure of multi-head self-attention

2.3 Trend Prediction Based on Transformer

The Transformer utilized in this study adopts the reference [12, 29] and modified for the current task. Transformer is a neural network model based on the self-attention mechanism. It can not only capture long-distance dependencies in sequence data, but can also be applied to the classification task of two-dimensional matrix data.

The fundamental concept of Transformer network is to use self-attention mechanism to consider the relationship between different positions in the sequence without the need for stepwise iteration like RNN. This section provides a detailed introduction to the architecture and main modules of Transformer, with the basic architecture of Transformer encoder briefly shown in Fig. 5a.

2.3.1 Embedding and Positional Encoding

In Transformer, the input data are first embedded into vector representations, which can capture the semantic information of different features. Additionally, to ensure that the network can identify positional relationships between elements, positional encodings are added to the embedding vectors and used together with embeddings to incorporate positional information of the input sequence into the model representation. Transformer in "Attention Is All You Need" [12] uses the commonly sine and cosine position encoding. The calculation formula is as follows:

$$\begin{aligned} PE(pos, 2i) &= \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \\ PE(pos, 2i+1) &= \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \end{aligned} \tag{2}$$

where pos represents the absolute position in the sequence, starting from 0 and incrementing; d_{model} is the dimension of the embedded word vector, usually equal to the embedding dimension in Transformer; $2i$ and $2i+1$ represent parity, and i represents the dimension in the word vector.

2.3.2 Encoder

Transformer typically utilize an encoder–decoder architecture. The encoder maps input sequences to a representation space, while the decoder generates output sequences from this space. The encoder consists of M stacked encoders with the same structure,as shown in Fig. 5a, each containing multi-head self-attention and fully connected networks, utilizing residual connections and normalization to enhance performance. In MPDTransformer, we only utilize the encoder segment of Transformer.

2.3.3 Self-attention

Self-attention mechanism is the core of Transformer, with the basic idea of comparing each element in a sequence with all other elements to calculate their correlation scores, and then using these scores as weights to blend information from other elements. The two modules on Fig. 5b and c illustrate the structure of self-attention mechanism. The basic definition of self-attention mechanism is as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V \tag{3}$$

where Q, K and V are three key matrices: Query matrix (Query), Key matrix (Key) and Value matrix (Value), obtained through linear transformations from the original input.

The multihead self-attention mechanism introduces multiple attention heads on this basis, with each head learning feature information in different representation spaces, thereby enhancing MPDTransformer's expressive power and generalization performance. The basic idea is to execute the attention function using projected versions of the query, key, and value matrices, and then connect the outputs of all attention functions together to generate the final result through a linear layer. The module on the Fig. 5d is the basic structure of the multi-head self-attention mechanism. Formula expression for multi-head self-attention is as follows:

$$\begin{aligned} &MultiHead(Q, K, V) = Concat(head_1, head_2, \ldots, head_h)W^O, \\ &head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{4}$$

where $i = 1, \ldots, h$, W_i^Q, W_i^K, W_i^V are projections of parameter matrices in multiple subspaces.

3 Experiment

In this section, we focus on the experimental analysis of MPDTransformer, detailing the experimental design and execution process to validate its effectiveness. We introduce the experimental dataset and data preprocessing methods in Sect. 3.1. Section 3.2 compares the predictive performance of different methods to assess the effectiveness of MPDTransformer. Section 3.3 provides an in-depth analysis of the individual components of MPDTransformer through ablation studies.

3.1 Datasets and Data Preprocessing

The trading data of 80 stocks from the Shenzhen Stock Exchange are used in our experiment. The dataset was obtained from Tushare, a financial data platform that provides comprehensive stock market information. We acquired the data via the API interface provided by Tushare. The dataset encompasses daily trading activities from April 2019 to April 2023, spanning multiple market cycles to capture diverse market conditions and enhance the representativeness of our analysis. For each stock, the dataset includes a wide range of daily trading information, such as open, high, low, close, pre-close, change, percentage change, trading volume, and trading amount. These features collectively provide a robust foundation for our research. Since the overall price range of each stock is different, we preprocess the data as follows:

(a) Data Cleaning. We began with data cleaning to ensure the integrity and consistency of the dataset. This step involved handling missing values and removing duplicates to prepare the data for further analysis.

(b) Data Normalization. We normalize the data using the maximum–minimum normalization method. This technique scales the data to the range [0,1], effectively eliminating the differences in magnitudes among various stocks. The formula used is:

$$x_{norm} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{5}$$

where x_{norm} is the normalized data point, x is the original data point, $x_{\min}$ is the minimum value in the dataset, and $x_{\max}$ is the maximum value in the dataset.

(c) Selection of technical indicators. Following normalization, we selected and calculated relevant technical indicators. These indicators were chosen based on their ability to capture key market dynamics and provide additional insights for our analysis.

In the financial stock market, technical indicators are crucial tools for analyzing market behavior and predicting price movements. These indicators, derived from historical data, provide actionable insights to guide decision-making. They are typically categorized into trend indicators, volatility indicators, and momentum indicators, each designed to capture specific market dynamics and provide unique insights. Given their importance, we carefully selected a set of technical indicators based on their relevance to our experiment objectives and their ability to comprehensively represent market conditions.

Trend Indicators. These indicators are designed to identify and follow market trends, helping to determine the overall direction of the market. We selected: Moving Average (MA), Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD), Average Directional Index(ADX).

Volatility indicators. These indicators are used to assess market conditions by measuring price volatility and identifying potential overbought or oversold conditions. We included: Random Indicators(KD), Relative Strength Index(RSI), Commodity Channel Index (CCI), Williams R (WR).

Momentum indicators. These indicators analyze the relationship between trading volume and price movements, providing insights into market strength and potential trend continuation. We selected: Bollinger Bands (BOLL), On-Balance Volume (OBV), Accumulation/Distribution Line (AD), Money Flow Index (MFI).

We selected the above indicators based on their ability to capture different aspects of market behavior and their complementary nature. Detailed descriptions of these indicators as shown in Table 4.

(d) Generation and labeling of two-dimensional images. Firstly, using the fusion method shown in Fig. 3a, we integrate the combined original trading data and technical indicators to generate the two-dimensional images. Subsequently, we utilize a sliding window of 28 × 31 with a stride of 1 to divide each sample into three categories: up, down, and flat, as shown in Fig. 3b. The labeling method is as follows:

$$label = \begin{cases} 1, close_t > close_{t-1} \\ 0, close_t = close_{t-1} \\ -1, close_t < close_{t-1} \end{cases} \tag{6}$$

where $close_{t-1}$ represents the stock close price for the day, $close_t$ represents the stock close price for the following day.

(d) Selection of evaluation metrics. In addition to the above processing, to asses the MPDTransformer's performation more accurately, Accuracy (Acc), Precision (Prec), Recall (Rec) and F1-score (F1) are used. Bigger values of these metrics indicate higher prediction accuracy. The corresponding discussion are listed in [16].

 https://doi.org/10.1007/s44196-025-00768-w

Table 1 Prediction performance of different methods

Method	Acc	Prec	Rec	F1
CNN	0.838	0.773	0.738	0.703
ResNet	0.794	0.750	0.794	0.726
Transformer	0.657	0.754	0.657	0.622
CNN[1]+CNN[2]	0.827	0.841	0.727	0.792
CNN+ResNet	0.580	0.438	0.406	0.361
CNN+Transformer	0.844	0.823	0.744	0.711
CAE+CNN	0.871	0.845	0.771	0.793
CAE+ResNet	0.907	0.864	0.707	0.793
CAE+Transformer*	**0.950**	**0.869**	**0.795**	**0.801**

[1] denotes the first CNN used for denoising [2] denotes the second CNN used for trend prediction * denotes the MPDTransformer we proposed

3.2 Predictive Performance of Different Methods

To evaluate the predictive performance of MPDTransformer, we compared the proposed method with several popular stock trend prediction models, including: CNN, ResNet, Transformer, CNN+CNN, CNN+ResNet, CNN+Transformer, CAE+CNN, CAE+ResNet, CAE+Transformer (Our proposed). All methods are tested on the same benchmark. Firstly, the acquired stock data was preprocessed, undergoing the same normalization and data cleaning operations. Secondly, we generate two-dimensional images, which were then subjected to sliding segmentation with a length set to 31. At the same time, we labeled the segmented window data for each stock, serving as the basis for classification.

In this experiment, to optimize model performance, we conducted a hyperparameter grid search experiment. The main parameters adjusted included the learning rate and the type of optimizer. Specifically, the learning rate was set to three different values (0.001, 0.0005, and 0.0001), and the optimizers selected were Adam and SGD, two classic methods. By training the model on the training set and evaluating its performance on the validation set, we recorded the model's performance under different hyperparameter combinations. When the learning rate was 0.0001 and the optimizer was Adam, the model achieved the highest accuracy of 95 percent. This indicates that an appropriate combination of learning rate and optimizer is crucial for the convergence speed and final performance of model training.

Table 1 presents the predictive performance of different methods, where the rows represent various prediction methods and the columns represent four evaluations. Bold numbers indicate the maximum values, representing the best performance.

From Table 1, we can make the following observations: (1) Models incorporating denoising techniques, such as those utilizing CAE and CNN, generally achieve higher evaluation metrics compared to those without denoising. It is evident that CAE network effectively reduces noise in the dataset, providing cleaner input data that enhances predictive accuracy. Notably, CAE outperforms CNN in terms of denoising effectiveness. (2) Models using Transformer for predictions demonstrate higher accuracy than those using CNN and ResNet. This indicates that Transformer performs better than those based on CNN and ResNet, showing its ability to capture long-term dependencies in the data. This suggests that Transformer is more effective in understanding and predicting complex market trends. (3) The technical indicators employed in our approach provide a better explanation of market conditions, further enhancing the model's performance.

3.3 Component Evaluation

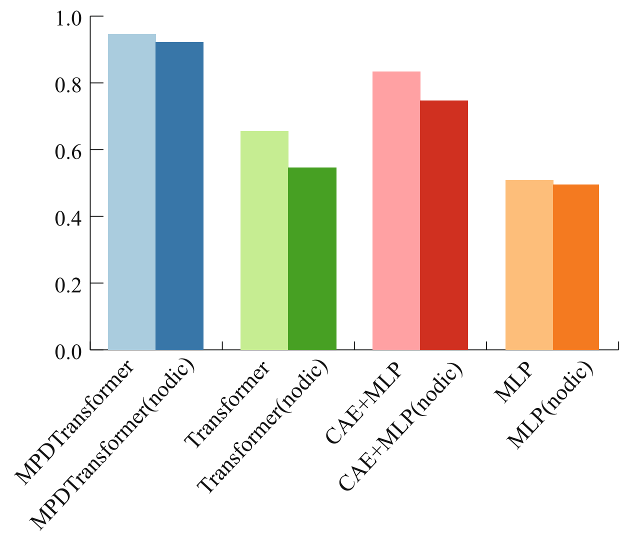
This section evaluates the contributions of various components in MPDTransformer through ablation experiments, providing a deeper analysis of their impact on overall performance. We will systematically remove different parts

Table 2 Ablation Experiment Results

Method	Technical Indicators	CAE Denoising	Transformer Predicting	Acc	Prec	Rec	F1
MPDTransformer	✓	✓	✓	**0.950**	**0.869**	**0.795**	**0.801**
MPDTransformer(nodic)		✓	✓	0.922	0.805	0.588	0.679
Transformer	✓		✓	0.656	0.754	0.656	0.623
CAE+MLP	✓	✓		0.834	0.549	0.726	0.792
Transformer(nodic)			✓	0.546	0.704	0.513	0.371
CAE+MLP(nodic)		✓		0.747	0.785	0.747	0.713
MLP	✓			0.509	0.595	0.509	0.378
MLP(nodic)				0.495	0.750	0.494	0.326

(nodic) denotes the model dataset without technical indicators

Fig. 6 Accuracy of different models



of MPDTransformer and analyze their effects on predictive performance, allowing for a more comprehensive understanding of the MPDTransformer's function. The results of these experiments will be organized in Table 2. Table 2 presents the results of the ablation experiments, where the rows represent various prediction methods and the columns indicate the components used by the corresponding methods and the evaluation metrics. Bold numbers indicate the maximum values, representing the best performance.

3.3.1 Impact of Technical Indicators

From Table 2, it can be observed that the performance of MPDTransformer(nodic) has decreased compared to MPDTransformer. MPDTransformer achieves an accuracy of 95.0%, while the accuracy of MPDTransformer without technical indicators is 92.2%, indicating an overall decline in performance. This suggests that technical indicators play an indispensable role in the model and significantly enhance its predictive capability.

Observing Fig. 6, it can be seen that MPDTransformer exhibits higher accuracy and stability in predicting stock trends when using technical indicators, compared to the control group without using technical indicators. This indicates that technical indicators help MPDTransformer better capture the rules of the stock market, enhancing the reliability of predictions. As an additional source of information, technical indicators enrich the feature space, enabling a more comprehensive understanding of market dynamics. Therefore, the conclusion is that technical indicators play an important role in stock trend prediction, effectively enchancing the performance and effectiveness.

To provide a more intuitive demonstration of the impact of different technical indicators, we constructed a heatmap (Fig. 7) that reflects the contribution of technical indicators from 6 randomly selected stocks to the

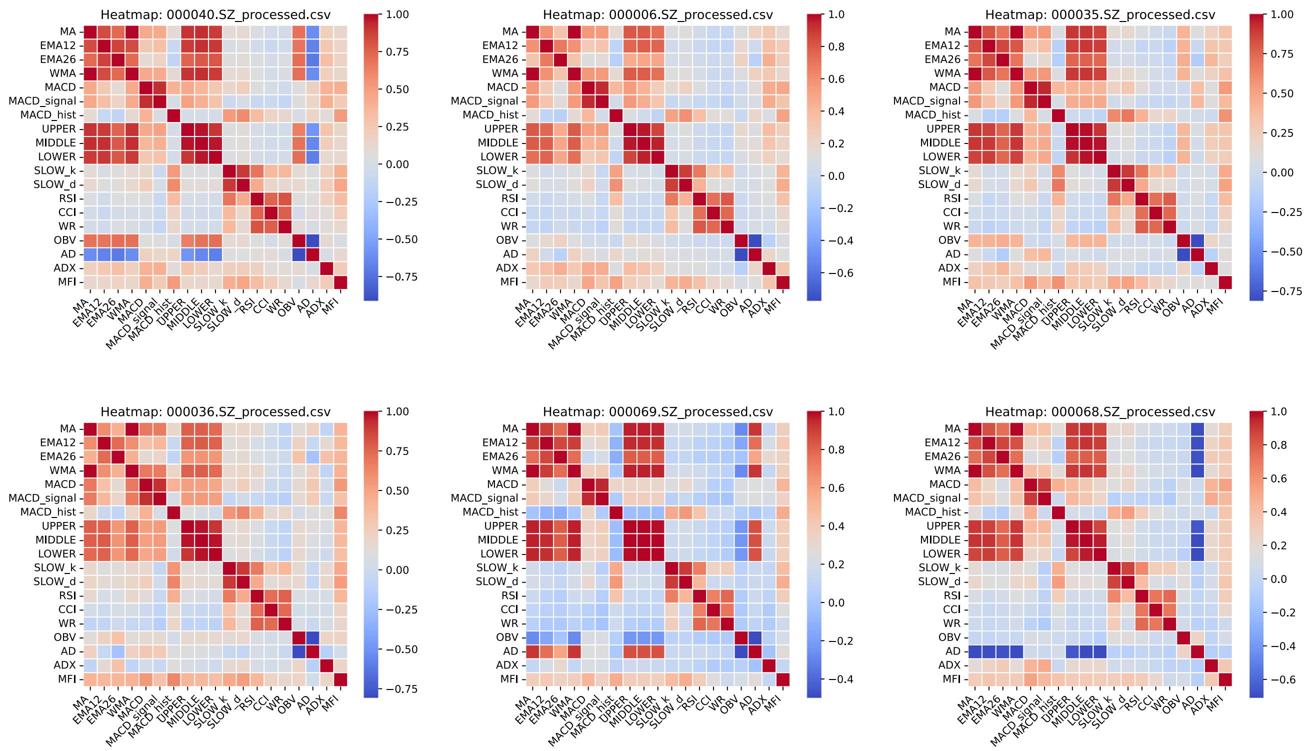


Fig. 7 Heatmap of technical indicators

MPDTransformer's performance, with color intensity representing changes in performance metrics. The heatmap clearly shows the significant role of key technical indicators in MPDTransformer. Similarly, other models also demonstrate that those with technical indicators perform better than their counterparts without technical indicators. Therefore, we conclude that technical indicators are crucial for improving the predictive accuracy of MPDTransformer.

3.3.2 Impact of Denoising

This section focuses on analyzing the impact of CAE denoising step to assess its effectiveness in improving data purity and MPDTranformer performance. The denoising process is crucial for the quality of the input data, as the presence of noise can interfere with the model's learning and affect prediction accuracy.

In Table 2, we compare the performance of MPDTransformer and Transformer. It is evident that the model with CAE denoising performs better than the model without CAE denoising. This conclusion is further supported by other experiments, such as CAE+MLP compared to MLP, MPDTransformer(nodic) compared to Transformer(nodic). By examining Fig. 8a, we can see that the evaluation metrics of MPDTransformer are higher than those of the model without denoising, indicating that data denoising enhances accuracy and stability during the training.

3.3.3 Impact of Transformer

This section explores the key role of Transformer in MPDTransformer, particularly its ability to capture complex temporal relationships and enhance classification accuracy. By comparing it with models that do not include Transformer, we quantitatively analyze the performance improvement provided by Transformer.

As shown in Table 2, the accuracy of MPDTransformer reaches 95.0%, while models using MLP instead of Transformer exhibit lower accuracy compared to their counterparts. This result highlights the importance of

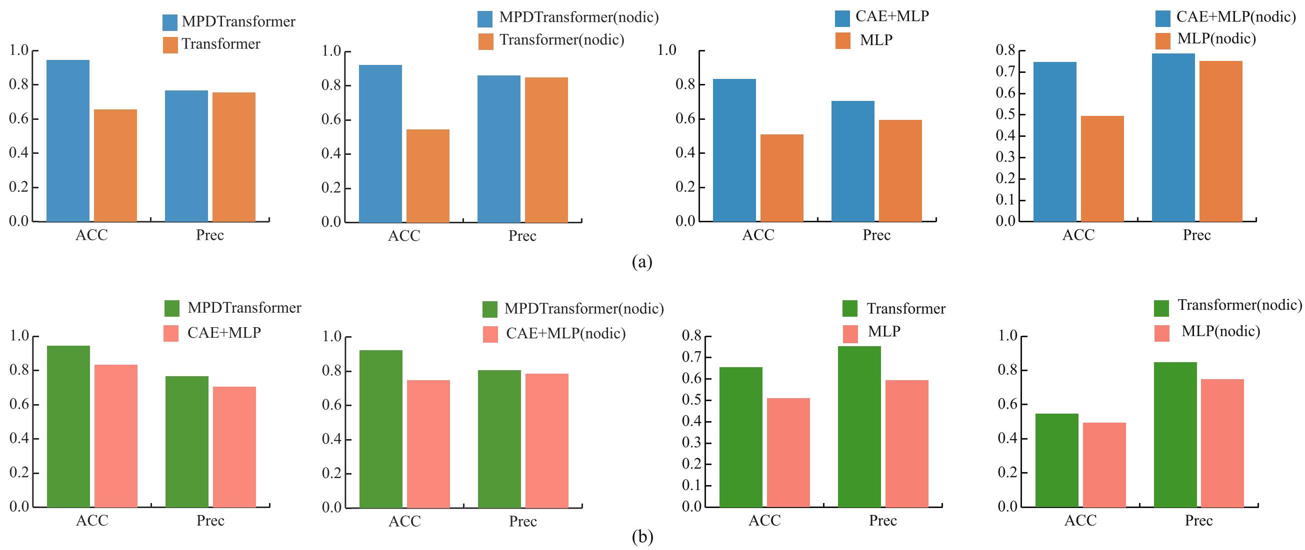


Fig. 8 Experimental results of different prediction methods **a** The impact of the denoising component on experimental results **b** The impact of Transformer component on experimental results

Transformer in capturing long-range dependencies in time series data, as MLP cannot fully leverage the advantages of Transformer. By observing Fig. 8b, it can be seen that the evaluation metrics of MPDTransformer is higher than that of other comparative experiments. This indicates that under the same denoising and processing conditions, the predictive performance of Transformer network is superior to that of other common prediction methods. It better captures the features and patterns within time series data, achieving higher performance levels and facilitating more accurate predictions.

The MPDTransformer's enhanced predictive performance is a direct result of the synergistic collaboration between its core components: technical indicators, CAE denoising, and the Transformer. Each component plays a unique role that, when combined, amplifies the model's ability to accurately forecast stock trends.

Firstly, the integration of technical indicators provides the model with a rich set of features that encapsulate various aspects of market behavior. These indicators offer a detailed view of market dynamics that would otherwise be challenging to discern. By incorporating these indicators, MPDTransformer gains the capacity to detect subtle patterns and trends, which are critical for making informed predictions.

Secondly, the CAE denoising is crucial for enchancing the quality of the input data. Noise in financial data can lead to misleading signals and reduce the model's effectiveness. The CAE step ensures that the data fed into the model is as pure as possible, thereby minimizing the noise. This results in a more stable and accurate model, as evidenced by the improved performance metrics when denoising is applied.

Lastly, the Transformer architecture adept at capturing the complex temporal relationships within the data. Unlike traditional models that may struggle with long-range dependencies, the Transformer excels at understanding how distant data points interact with each other. This ability is particularly valuable in time series analysis, where past events can significantly influence future outcomes.

In conclusion, the MPDTransformer's design is a testament to the power of synergy. The combination of insightful technical indicators, precise denoising, and the sophisticated Transformer architecture creates a model that not only surpasses its individual parts but also sets a new standard in stock trend prediction.

4 Market Trading Simulation

In the experimental section, we validated the performance of different models in predicting stock trends. To further assess the MPDTransformer's performance in a real market environment, we designed a market simulation trading

 https://doi.org/10.1007/s44196-025-00768-w

Table 3 Market trading simulation results

Stock	MPDTransformer		CNN	
	Final balance	Total profit	Final balance	Total profit
000012.SZ	280.97	**220.97**	98.73	38.73
000025.SZ	248.61	**188.61**	235.93	175.93
000037.SZ	62.83	52.83	134.55	**74.55**
000059.SZ	104.51	**44.51**	98.41	38.41
000063.SZ	196.09	**136.09**	126.74	66.74
000078.SZ	65.17	25.17	87.69	**27.69**
Average profit		**111.36**		70.34

experiment. By using MPDTransformer to predict future trends for multiple stocks and applying a simple trading strategy, we calculated the final account balance and profits for each stock to evaluate the MPDTransformer's actual profitability.

In the market simulation, we employed the following trading strategy: when MPDTransformer predicts down, we buy stocks with an amount of 1 RMB for each transaction; when MPDTransformer predicts up, we sell stocks worth 1 RMB; if the prediction indicates flat, we remain cautious and do not execute any buy or sell operations. The initial capital was set at 60 RMB. Throughout the trading process, we recorded each transaction's actions, stock holdings, and changes in account balance.

We collected historical data from 6 stocks listed on Shenzhen Stock Exchange, including open price, close price, highest price, lowest price, and trading volume. Additionally, we generated the technical indicators and combined them with the original trading data to create two-dimensional image data for MPDTransformer. MPDTransformer predicted future market trends based on the sliding window data to decide whether to buy, sell, or hold.

Simulated experiments were conducted using both MPDTransformer and CNN, and the results are shown in Table 3, it can be seen that MPDTransformer performs relatively consistently across multiple stocks. Table 3 shows a comparison of market simulations using MPDTransformer and CNN, where the rows represent multiple stock datasets and the columns indicate the performance comparison of the two methods for the corresponding stocks. Bold numbers denote the maximum values, representing the best performance. Compared to CNN, the average profit from simulated trading using MPDTransformer is higher, indicating its potential profitability in actual market environments. Then, we generate images showing the account balance changes and stock position changes of each stock, allowing for further comparison between the two models.

Figure 9 shows the changes in account balance for 4 stocks throughout the trading process. It can be seen that as trading progresses, the account balance exhibits a fluctuating trend, with stocks experiencing significant appreciation based on the MPDTransformer's predictions, highlighting the MPDTransformer's potential profitability in actual trading. It is also evident that at the end of the trading period, the value for MPDTransformer is higher than that of CNN, indicating that the profits from simulations using MPDTransformer are greater than those from CNN.

Through this market simulation trading experiment, we further validated the practical application potential of MPDTransformer. The model was able to generate profits in most cases, demonstrating its feasibility and effectiveness in actual trading. Future research will focus on exploring more complex trading strategies and testing them on larger dataset to further enhance the MPDTransformer's profitability and stability.

5 Conclusion

To improve the accuracy of stock trend prediction, we propose a multi-factor prediction model—MPDTransformer. The performance of MPDTransformer was experimentally verified by combining CAE feature extraction, Transformer predictive capabilities, and effective use of technical indicators. Experimental results demonstrate that

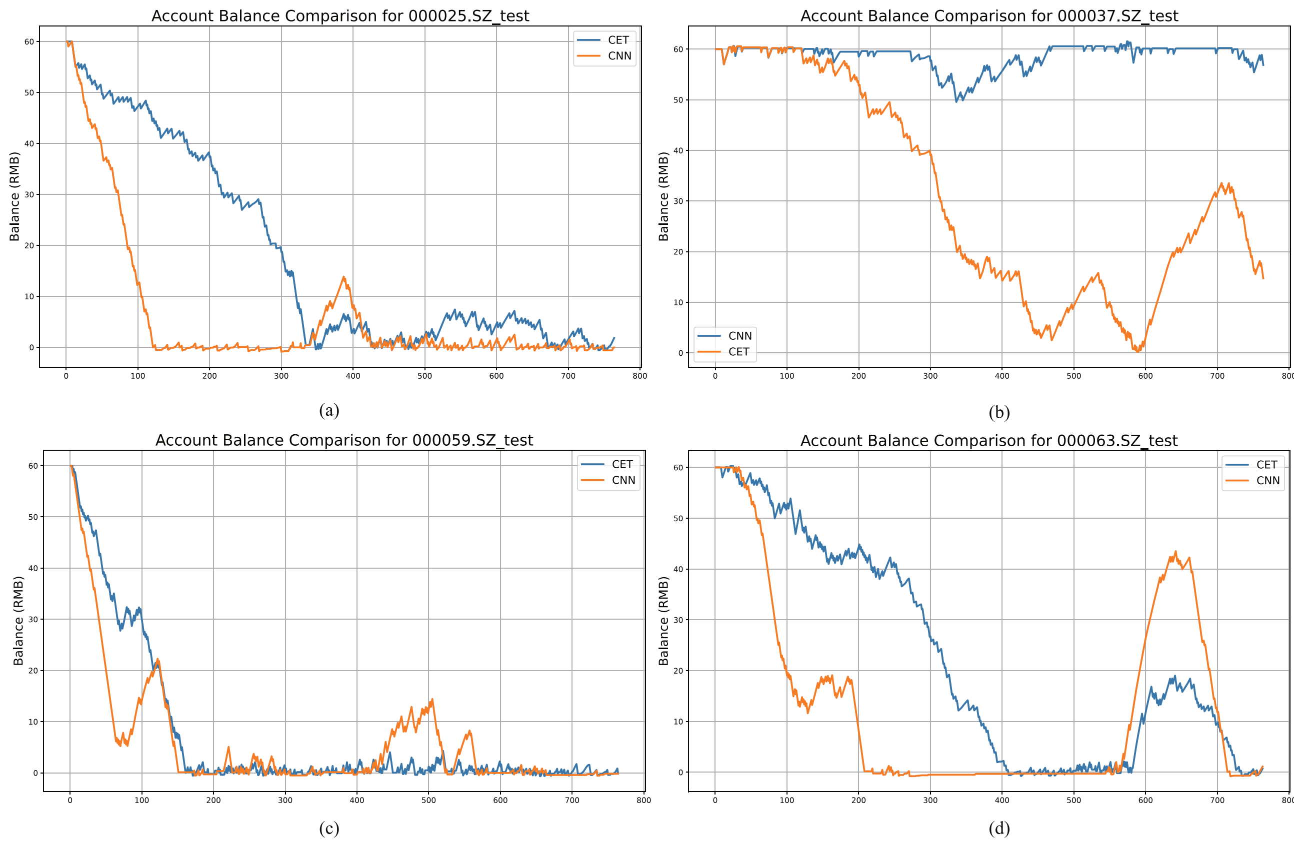


Fig. 9 Account balance change for stocks

MPDTransformer can provide highly accurate and stable prediction results, leading to the following conclusions: (1) Selecting comprehensive technical indicators is important in the domain of stock trend prediction, as it provides valuable information to MPDTransformer. Multi-factor data allow for a more holistic representation of stock market dynamics, which improves MPDTransformer's prediction accuracy. (2) Proper denoising processing allows MPDTransformer to better understand and process stock data, which improves its stability and accuracy. Experimental results show that using CAE for denoising has the best effect, effectively reducing noise and outliers in data, and improving MPDTransformer prediction. Although CAE is effective at denoising, it can sometimes remove important information along with the noise, potentially leading to minor misidentifications and loss of critical features needed for accurate predictions. Currently, this issue has minimal impact, but addressing it is essential for model robustness and reliability. In the future, we aim to optimize the CAE denoising process and minimize information loss while maintaining noise reduction effectiveness. (3) MPDTransformer outperforms traditional forecasting methods in stock trend forecasting. When processing time series data, Transformer can better focus on information at various points in the sequence and capture it. It can process multidimensional information input more naturally, increasing MPDTransformer's adaptability and flexibility while also achieving more accurate and stable prediction results.

In conclusion, a new method for stock market prediction has been developed, which improves the accuracy and stability of stock trend predictions. Our current approach primarily serves as a simple informational aid, utilizing CAE for denoising and Transformer for predicting future stock trends. While this method can provide valuable insights into market movements, it is intended as a basic tool for understanding general trends rather than a comprehensive basis for detailed trading decisions. Therefore, it should be used as a supplementary guide rather than a definitive strategy for buying or selling.

https://doi.org/10.1007/s44196-025-00768-w

In the future, we aim to refine our approach by exploring more sophisticated trading strategies. This includes predicting specific future price values or determining optimal buy/sell ratios to guide more complex trading decisions. By doing so, we hope to provide a more nuanced and actionable framework for investors and traders. In addition, given the computational complexity of Transformers and convolutional autoencoders, we will focus on improving the scalability of MPDTransformer when applied to larger datasets or across different stock markets. These efforts will not only address computational complexity challenges, but will also ensure MPDTransformer's continued effectiveness and efficiency in practical trading systems.

Appendix A Technical Indicator

See Table 4.

Table 4 Description of technical indicator variables

Technical indicator	Related description
Moving Average(MA)	The average close price over a period of time
Exponential Moving Average (EMA)	A weighted average of close prices
Moving Average Convergence Divergence (MACD)	An indicator of moving average convergence and divergence, indicating price trends
Bollinger Bands(BOLL)	An indicator of price volatility range
Random Indicators(KD)	Indicators used to identify overbought and oversold conditions
Relative Strength Index (RSI)	An indicator measuring upward and downward pressure on prices
Commodity Channel Index (CCI)	An indicator used to measure whether the stock market conforms to a normal distribution
Williams indicators(WR)	An indicator of the position of prices relative to the highest and lowest prices over a certain period
On-Balance Volume(OBV)	An indicator measuring buying and selling pressure
Accumulation/Distribution (AD)	An indicator measuring inflow and outflow of funds
Average Directional Index (ADX)	An indicator of trend strength and direction
Money Flow Index(MFI)	An indicator measuring the strength of money inflows and outflows

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (62176140), the Central Guidance on Local Science and Technology Development Fund of Shandong Province (YDZX2022093) and Technology Innovation Program of Yantai City (2023XDRH001).

Author Contributions Feng Zhao, Shuoru Chen:Conceptualization, Methodology. Shuoru Chen: Data curation, Writing original draft. Xiaoyan Qiao: Visualization, Investigation. Mingli Zhang, Caiming Zhang:Supervision. Xiliang Li: Reviewing and Editing.

Data availability No datasets were generated or analysed during the current study.

Declaratins

Conflict of interest The authors declare no competing interests.

References

1. Sezer, O.B., Ozbayoglu, A.M.: Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. Appl. Soft Comput. **70**, 525–538 (2018). https://doi.org/10.1016/j.asoc.2018.04.024
2. Jiang, M., Jia, L., Chen, Z., Chen, W.: The two-stage machine learning ensemble models for stock price prediction by combining mode decomposition, extreme learning machine and improved harmony search algorithm. Ann. Oper. Res. **309**(2), 553–585 (2022). https://doi.org/10.1007/s10479-020-03690-w
3. Singh, P., Dhiman, G.: A hybrid fuzzy time series forecasting model based on granular computing and bio-inspired optimization approaches. J. Comput. Sci. **27**, 370–385 (2018). https://doi.org/10.1016/j.jocs.2018.05.008
4. Zhang, J., Li, L., Chen, W.: Predicting stock price using two-stage machine learning techniques. Comput. Econ. **57**(4), 1237–1261 (2021). https://doi.org/10.1007/s10614-020-10013-5
5. Liu, Q., Tao, Z., Tse, Y., Wang, C.: Stock market prediction with deep learning: the case of china. Financ. Res. Lett. **46**, 102209 (2022). https://doi.org/10.1016/j.frl.2021.102209
6. Cagliero, L., Garza, P., Attanasio, G., Baralis, E.: Training ensembles of faceted classification models for quantitative stock trading. Computing **102**(5), 1213–1225 (2020). https://doi.org/10.1007/s00607-019-00776-7
7. Singh, S., Parmar, K.S., Kumar, J.: Soft computing model coupled with statistical models to estimate future of stock market. Neural Comput. Appl. **33**(13), 7629–7647 (2021). https://doi.org/10.1007/s00521-020-05506-1
8. Guang, L., Mao, Y., Sun, Q., Huang, H., Gao, W., Li, X., Shen, J., Li, R., Wang, X.: Multi-scale two-way deep neural network for stock trend prediction, pp. 4505–4511 (2020). https://doi.org/10.24963/ijcai.2020/621
9. Chen, Y., Diao, Y.: Effectiveness of and bp-neural network for overnight stock price forecasting: an example of csi 300. Front. Bus. Econ. Manag. **11**(1), 27–30 (2023). https://doi.org/10.54097/fbem.v11i1.11599
10. Khandelwal, I., Adhikari, R., Verma, G.: Time series forecasting using hybrid arima and ann models based on dwt decomposition. Proc. Comput. Sci. **48**, 173–179 (2015). https://doi.org/10.1016/j.procs.2015.04.167
11. Kim, K.-j: Financial time series forecasting using support vector machines. Neurocomputing **55**(1), 307–319 (2003). https://doi.org/10.1016/S0925-2312(03)00372-2
12. Hafezi, R., Shahrabi, J., Hadavandi, E.: A bat-neural network multi-agent system (bnnmas) for stock price prediction: Case study of dax stock price. Appl. Soft Comput. **29**, 196–210 (2015). https://doi.org/10.1016/j.asoc.2014.12.028
13. Xiong, R., Nichols, E.P., Shen, Y.: Deep learning stock volatility with google domestic trends. arXiv (2016)
14. Fischer, T., Krauss, C.: Deep learning with long short-term memory networks for financial market predictions. Eur. J. Oper. Res. **270**(2), 654–669 (2018). https://doi.org/10.1016/j.ejor.2017.11.054
15. Liu, H., Long, Z.: An improved deep learning model for predicting stock market price time series. Digit. Signal Process. **102**, 102741 (2020). https://doi.org/10.1016/j.dsp.2020.102741
16. Chen, W., Jiang, M., Zhang, W.-G., Chen, Z.: A novel graph convolutional feature based convolutional neural network for stock trend prediction. Inf. Sci. **556**, 67–94 (2021). https://doi.org/10.1016/j.ins.2020.12.068
17. Yang, Z., Zhao, T., Wang, S., Li, X.: Mdf-dmc: a stock prediction model combining multi-view stock data features with dynamic market correlation information. Expert Syst. Appl. **238**, 122134 (2024). https://doi.org/10.1016/j.eswa.2023.122134
18. Long, W., Lu, Z., Cui, L.: Deep learning-based feature engineering for stock price movement prediction. Knowl.-Based Syst. **164**, 163–173 (2019). https://doi.org/10.1016/j.knosys.2018.10.034
19. Wu, J.M.-T., Li, Z., Herencsar, N., Vo, B., Lin, J.C.-W.: A graph-based cnn-lstm stock price prediction algorithm with leading indicators. Multimedia Syst. **29**(3), 1751–1770 (2023). https://doi.org/10.1007/s00530-021-00758-w
20. Lin, H., Zhao, J., Liang, S., Kang, H.: Prediction model for stock price trend based on convolution neural network. J. Intell. Fuzzy Syst. **39**(4), 4999–5008 (2020). https://doi.org/10.3233/JIFS-179985
21. Gao, T., Chai, Y.: Improving stock closing price prediction using recurrent neural network and technical indicators. Neural Comput. **30**(10), 2833–2854 (2018). https://doi.org/10.1162/neco_a_01124
22. Carosia, A.E.D.O., da Silva, A.E.A., Coelho, G.P.: Predicting the brazilian stock market with sentiment analysis, technical indicators and stock prices: a deep learning approach. Comput. Econ. (2024). https://doi.org/10.1007/s10614-024-10636-y
23. Wang, Z., Hu, Z., Li, F., Ho, S.-B., Cambria, E.: Learning-based stock trending prediction by incorporating technical indicators and social media sentiment. Cogn. Comput. **15**(3), 1092–1102 (2023). https://doi.org/10.1007/s12559-023-10125-8
24. Zhao, Y., Yang, G.: Deep learning-based integrated framework for stock price movement prediction. Appl. Soft Comput. **133**, 109921 (2023). https://doi.org/10.1016/j.asoc.2022.109921

25. Jia, P., Zhang, M., Shen, Y.: Deep spectral unmixing framework via 3d denoising convolutional autoencoder. IET Image Proc. **15**(7), 1399–1409 (2021). https://doi.org/10.1049/ipr2.12113
26. Emami, H., Dang, X.-H., Shah, Y., Zerfos, P.: Modality-aware transformer for financial time series forecasting. arXiv (2024). https://doi.org/10.48550/arXiv.2310.01232
27. Leung, M.-F., Wang, J.: Cardinality-constrained portfolio selection based on collaborative neurodynamic optimization. Neural Netw. **145**, 68–79 (2022). https://doi.org/10.1016/j.neunet.2021.10.007
28. Xie, L., Yu, S.: Unsupervised feature extraction with convolutional autoencoder with application to daily stock market prediction. Concurr. Comput. Pract. Exp. **33**(16), 6282 (2021). https://doi.org/10.1002/cpe.6282
29. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv (2021). https://doi.org/10.48550/arXiv.2010.11929

Authors and Affiliations

Xiliang Li[1] · Shuoru Chen[2] · Xiaoyan Qiao[1,3,4] · Mingli Zhang[2] · Caiming Zhang[2] · Feng Zhao[2,3,4]

✉ Feng Zhao
zhaofeng1016@126.com

Xiliang Li
lixiliang@amss.ac.cn

Shuoru Chen
csr000412@163.com

Xiaoyan Qiao
qiaoxy1026@126.com

Mingli Zhang
zhangmellie@gmail.com

Caiming Zhang
czhang@sdu.edu.cn

[1] School of Mathematics and Information Science, Shandong Technology and Business University, Yantai 264005, China

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

[3] Yantai Key Laboratory of Big Data Modeling and Intelligent Computing, Yantai 264005, China

[4] Immersion Technology and Evaluation Shandong Engineering Research Center, Yantai 264005, China