Stock Price Prediction Using BiLSTM and a Modified Transformer

WANG SHUZHEN[1,2]
[1]School of Information Science and Technology, Tan Kah Kee College, Xiamen University, Zhangzhou 363105, Fujian, China
[2]Hongwang Laboratory, School of Information Science and Technology, Xiamen University, Zhangzhou 363105, Fujian, China

ABSTRACT Stock price prediction remains a fundamental problem in financial forecasting. To improve predictive accuracy and stability, this paper proposes a bidirectional long short-term memory-modified Transformer-temporal convolutional network (BiLSTM-MTRAN-TCN). The Transformer is first redesigned and combined with a temporal convolutional network (TCN) to form a modified Transformer-temporal convolutional network (MTRAN-TCN). The resulting module is then integrated with a bidirectional long short-term memory network (BiLSTM) to construct a hybrid forecasting architecture. The proposed model combines the global dependency modeling capability of the Transformer, the bidirectional temporal feature extraction ability of BiLSTM, and the sequence modeling and generalization advantages of TCN. Experiments on five stock indices and 14 A-share stocks are conducted to evaluate both the modified Transformer and the contribution of the BiLSTM component. The proposed model achieves the best average performance in the stock-index experiments, attains the highest R^2 on 85.7% of the stock datasets, and yields the lowest RMSE on 78.6% of the stock datasets. Its performance also remains relatively stable across different time periods. These results indicate that BiLSTM-MTRAN-TCN offers strong accuracy, stability, and generalization ability for stock price prediction.

INDEX TERMS stock price prediction; BiLSTM; modified Transformer; temporal convolutional network; deep learning

I. INTRODUCTION

Stock price prediction has attracted sustained attention from investors and researchers because of its theoretical significance and practical value [1]. With the continuous development of capital markets, investors have become increasingly concerned with stock price fluctuations and trend changes [2]. For economists and market participants, identifying stock price trends in advance is an important task [3], [4], as it can support better investment decisions and potentially improve returns. However, stock markets are highly volatile and are affected by random noise and many other factors, which makes price dynamics complex and difficult to predict [5]. Consequently, extracting potentially predictable information from financial time series remains a central issue in stock price forecasting.

Early studies mainly relied on traditional time-series models, among which the autoregressive integrated moving average (ARIMA) model is representative [6]. Later, Babu and Reddy [7] applied a combined ARIMA-generalized autoregressive conditional heteroskedasticity (GARCH) model to data from the Indian NSE market. Bayesian vector autoregression, Kalman filtering, and related methods have also been explored in this area. In general, these approaches can be effective for short-term forecasting, but they are less suitable for complex nonlinear problems and often exhibit limited long-term predictive capability [8]. To address these limitations, machine learning methods were gradually introduced into stock prediction tasks [2], [9], [10]. Support vector machines (SVMs), decision trees, naive Bayes classifiers, and random forests have shown advantages in handling complex features and large-scale data [9]. For example, Wang et al. [11] used a combined decision tree and SVM model to predict future price movements; Chen et al. [12] proposed a feature-weighted SVM combined with a k-nearest neighbor model for stock market index prediction and reported favorable short-, medium-, and long-term results; Yan et al. [13] further proposed a hybrid model based on Pearson correlation and random forests for short-term stock price regression.

With the rapid development of deep learning, researchers have increasingly adopted data-driven methods for stock forecasting rather than relying heavily on prior domain knowledge. Deep learning has therefore become an important research direction in this field. Representative models include gated recurrent units (GRUs), recurrent neural networks (RNNs), convolutional neural networks (CNNs), long short-term memory networks (LSTMs), and bidirectional long short-term memory networks (BiLSTMs). In 2017 and 2018, wavelet neural networks and CNNs were successively applied to stock prediction [14], [15], and the corresponding results suggested that CNNs can extract informative features for time-series forecasting. In 2018, a one-dimensional convolution-LSTM (Conv1D-LSTM) model was proposed to combine the feature extraction ability of CNNs with the sequence modeling ability of LSTMs, and its predictive performance was reported to exceed that of traditional machine learning models [16]. In 2019, Yang and Wang [17] used BiLSTM to forecast global stock indices and found that BiLSTM achieved high prediction accuracy and strong generalization. In 2020, an LSTM regression model was applied to forecasting the Indian NIFTY 50 index, and deep learning models were reported to outperform traditional machine learning methods [10].

In recent years, attention mechanisms have gradually become an important component in financial time-series forecasting because they emphasize the information that has greater influence on prediction results. In 2021, Lu et al. [8] introduced an attention mechanism based on CNN and BiLSTM into stock price prediction and proposed a CNN-BiLSTM-AM model, which outperformed several existing methods. In 2022, another study combined LSTM with an attention mechanism and used wavelet transforms to denoise historical stock data [18]. Meanwhile, the Transformer, as a representative attention-based architecture, has provided a new perspective for sequence modeling [19]. Researchers have also begun to apply Transformer-based models to stock trend forecasting. Ding et al. [20] showed in 2020 that a Transformer enhanced with multi-Gaussian priors could be used for stock movement prediction. In 2021, a self-attention-based Transformer network was proposed and validated on electricity consumption and traffic datasets for time-series forecasting [21]. In 2022, Zhang et al. [5] proposed a Transformer encoder-based attention network that fused media text and stock price information to predict stock movements effectively. Peng et al. [22] combined LSTM and Transformer to predict Chinese banking stock prices, while Wang et al. [23] used a Transformer model for stock market index forecasting and reported better performance than other classical methods.

Despite these advances, several limitations remain in the literature. First, many Transformer-based studies focus on predicting price direction rather than direct numerical forecasting of stock prices. Second, many studies conduct experiments on only a single stock or a single stock index, which limits the breadth of empirical validation. Third, some models exhibit unstable performance when the time span changes, suggesting potential temporal sensitivity. Therefore, there is still room for improvement in model design, prediction accuracy, and generalization ability in stock forecasting research.

To improve the stability and accuracy of stock price prediction, this paper proposes a Transformer-based method termed BiLSTM-MTRAN-TCN. The method is applicable to both stock-index forecasting and individual stock forecasting. Specifically, BiLSTM and TCN are incorporated into a Transformer encoder-based framework to construct a hybrid prediction network. The Transformer provides strong global feature modeling capability, BiLSTM captures bidirectional temporal information, and TCN enhances sequence dependency modeling and generalization.

The main contributions of this paper are as follows:

- A modified Transformer with an introduced TCN is constructed to form the MTRAN-TCN model, making the Transformer more suitable for stock price forecasting.
- The proposed framework combines BiLSTM and MTRAN-TCN, thereby integrating the structural advantages of BiLSTM, Transformer, and TCN.
- A bidirectional stock selection strategy is proposed to choose experimental stocks.
- The proposed model is compared with representative methods from the literature to verify its effectiveness.
- Experimental results show that the proposed model exhibits favorable generalization ability and stable predictive performance over the considered time spans.

II. METHODOLOGICAL BACKGROUND

A. TRANSFORMER

The Transformer, proposed by Google in 2017, is a classical model in natural language processing (NLP) [19]. Widely used models such as BERT are also built on the Transformer architecture. By relying on self-attention rather than recurrent computation, the Transformer models global dependencies within a sequence and offers strong parallel computing capability. Compared with conventional recurrent networks, it can capture long-range dependencies more effectively and alleviate gradient vanishing to some extent.

The Transformer consists of an encoder and a decoder. As shown in Fig. 1, the encoder is formed by stacking multiple encoder layers and is used to encode the input sequence, while the decoder generates the final output based on the encoded features. The most critical component in the encoder is the multi-head self-attention mechanism, which allows the model to attend to information at different positions simultaneously and thus capture both short-term and long-term dependency patterns. The output of self-attention is computed from the matrices Q, K, and V, where d_k denotes the dimensionality of the key vector:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

Because of its strong contextual modeling capability, the Transformer is also well suited to time-series forecasting. However, the original Transformer was designed primarily for machine translation and cannot be directly applied to stock price prediction without modification. To adapt it to stock time-series modeling, the encoder component is retained as the core structure and modified accordingly, as illustrated in Fig. 3.

B. TCN

TCN is a neural network architecture designed for time-series forecasting and was first proposed by Lea et al. in 2016 [25]. Its architecture mainly consists of causal convolutions, dilated convolutions, and residual connections. Each convolution layer adopts a unidirectional causal structure, as shown in Fig. 2. Causal convolution ensures that the output at time step T depends only on information at time T and earlier, thereby preventing future information leakage. Moreover, this structure can accept input sequences of arbitrary length and produce outputs of the same length. The causal convolution is defined as follows:

$$F(s) = \sum_{i=0}^{k-1} f(i) x_{s-di} \tag{2}$$

Dilated convolution expands the receptive field without substantially increasing the number of network layers or introducing pooling operations, making it more effective for capturing long-range dependencies.

III. THE PROPOSED BILSTM-MTRAN-TCN METHOD

MTRAN-TCN is an improved model obtained by introducing TCN into the Transformer. This section first describes how the Transformer is modified and then presents the overall architecture of the proposed method.

A. FROM TRANSFORMER TO MTRAN-TCN

To make the Transformer more suitable for stock price prediction, the decoder-side design is modified as shown in Fig. 3.

- The Input Embedding module is removed (see Fig. 1). This module is mainly used for textual vector representation in machine translation, whereas stock forecasting does not involve this type of discrete text input.
- The Position Encoding module is moved out of MTRAN-TCN and placed before the BiLSTM module, as shown in Fig. 4.
- The original Transformer decoder is replaced by TCN layers, a fully connected layer, and a hyperbolic tangent (Tanh) function.
- Additional decoder inputs are removed, and only the encoder output is retained as the input to subsequent modules.

Existing studies have shown that TCN performs well in sequence prediction tasks. In 2022, Chen et al. [26] used TCN to extract time-series features from traffic datasets. Wang et al. [27] proposed a multivariate TCN-Attention model for daily traffic volume forecasting. In 2023, Yang et al. [28] proposed a bidirectional short-term memory network integrating self-attention and TCN for stock forecasting. These studies suggest that TCN is well suited to sequence modeling and can deliver competitive predictive performance. Accordingly, TCN is introduced into the Transformer in this work to strengthen temporal modeling capability.

The improved model is denoted as MTRAN-TCN, and its internal structure is shown in Fig. 3. The left half of the model is the Transformer encoder, and the right half consists of TCN and a fully connected layer. The Transformer encoder is built by stacking multiple encoder layers, each of which contains two sublayers: the first includes multi-head attention, normalization, and a residual connection, and the second includes a feed-forward fully connected layer, normalization, and a residual connection. The TCN is composed of multiple residual blocks, each of which mainly contains two dilated causal convolution layers, WeightNorm, and Dropout. The output of the encoder is used as the input to the first residual block of the TCN.

B. NETWORK ARCHITECTURE

TCN is introduced into the Transformer to better adapt the architecture to stock time-series forecasting. Although the Transformer offers strong parallel computation and global information extraction, it remains less effective in modeling local temporal dependencies. TCN can capture both high-level and low-level temporal features while maintaining stable gradients, thereby improving training efficiency and predictive accuracy. Its introduction allows the complementary strengths of Transformer and TCN in global modeling and sequence dependency modeling to be exploited more effectively. In addition, BiLSTM has a clear advantage in capturing bidirectional temporal signals. BiLSTM is therefore combined with the modified Transformer to construct the hybrid BiLSTM-MTRAN-TCN network shown in Fig. 4.

At each time step, the output of BiLSTM is influenced by both the current input and the historical memory state, enabling it to characterize bidirectional dependencies within the sequence. However, stock prediction often involves long input sequences, and BiLSTM may still suffer from gradient attenuation, which can weaken the learning of salient features. By contrast, the multi-head self-attention mechanism in MTRAN-TCN can emphasize important features, suppress irrelevant information, and improve predictive accuracy, but its ability to encode sequential order remains limited. Although positional encoding models positional information through sine and cosine functions, it cannot fully compensate for the absence of explicit temporal dependency modeling. Therefore, BiLSTM is used before the multi-head self-attention module of MTRAN-TCN to strengthen sequential dependency modeling and improve overall predictive performance.

As the number of network layers and training epochs increases, model degradation may occur and performance on new data may become unstable. To address this issue, TCN layers are introduced to handle variable-length time series. The convolutional structure models sequence dependencies, while residual connections alleviate the optimization difficulties of deep networks, thereby improving both generalization ability and training efficiency.

Consequently, by introducing TCN into the Transformer and further integrating it with BiLSTM, the proposed hybrid network can effectively improve representation ability and training efficiency.

In the proposed framework, stock trading data and technical indicators are used as inputs (see Section IV-A), and the next-day closing price is used as the output. The model input is a three-dimensional tensor defined by the number of samples, time steps, and features. The input first passes through a positional encoding layer, then through BiLSTM for temporal feature extraction, and is subsequently fed into the Transformer encoder. The TCN layer further refines the extracted features, and the final output is generated by a fully connected layer followed by an activation function. As shown in Fig. 4, the model consists primarily of a positional encoding layer, a BiLSTM layer, Transformer encoder layers, TCN layers, and a dense layer.

IV. EXPERIMENTAL SETUP

This section introduces the experimental datasets, evaluation metrics, and network parameter settings.

A. DATASETS

Previous studies [29]-[32] often selected only a small number of stocks from the Shanghai and Shenzhen markets or related indices. The selected samples were usually relatively stable and exhibited limited volatility, resulting in limited experimental coverage and less persuasive model conclusions.

To broaden experimental coverage, five representative stock indices and 14 A-share stocks are selected as research objects. The stock indices include the A-Share Index, the Shanghai Composite Index, the Shenzhen Component Index, the CSI 300 Index, and the ChiNext Index, as listed in Table 1.

Table 1. Selected stock indices

No.	Index name	Index code
1	A-Share Index	000002.XSHG
2	Shanghai Composite Index	000001.XSHG
3	Shenzhen Component Index	399001.XSHE
4	CSI 300 Index	399300.XSHE
5	ChiNext Index	399006.XSHE

For individual stock selection in the Shanghai and Shenzhen markets, a bidirectional stock selection strategy is adopted to further expand experimental coverage. Horizontally, the samples are divided into large-cap and small-cap stocks according to company market capitalization. Vertically, seven industry categories are selected: finance, real estate, coal, steel, nonferrous metals, petrochemicals, and automobiles. The selected stock samples are listed in Table 2.

Table 2. Selected A-share stocks

Category	Large-cap stock	Small-cap stock
Finance	China Merchants Bank (SH600036)	Sinolink Securities (SH600109)
Real estate	Poly Developments (SH600048)	Tibet Urban Development (SH600773)
Coal	China Shenhua (SH601088)	Power Investment Energy (SZ002128)
Steel	CITIC Special Steel (SZ000708)	Fangda Carbon (SH600516)
Nonferrous metals	Tianqi Lithium (SZ002466)	Yunnan Copper (SZ000878)
Petrochemicals	PetroChina (SH601857)	Yueyang Xingchang (SZ300164)
Automobiles	BYD (SZ002594)	Dongfeng Automobile (SH600006)

The stock index data were obtained from JoinQuant (https://www.joinquant.com/research), and the data for the 14 A-share stocks were obtained from Tushare (https://tushare.pro/).

For each stock, the most recent 2700 trading days were selected, covering the period from February 2012 to May 2023. For stock indices, the historical trading data were used as features, including closing price, highest price, lowest price, opening price, percentage change, price change, turnover rate, and trading volume. For individual stocks, the first six fields were used, and two technical indicators were additionally introduced, namely the 5-day moving average and the 10-day moving average.

Because the features in stock datasets have different scales, the raw data must be normalized [33]. Z-score standardization is adopted, which is defined as

$$y_i = \frac{x_i - \bar{x}}{s} \tag{3}$$

where y_i denotes the standardized value, x_i denotes the input data, $\bar{x}$ denotes the mean of the input data, and s denotes the standard deviation of the input data.

B. PERFORMANCE EVALUATION

The mean squared error (MSE), mean absolute error (MAE), root mean square error (RMSE), and coefficient of determination (R^2) are used as evaluation metrics. These metrics are defined as follows:

$$\left.\begin{aligned} \mathrm{MSE} &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \\ \mathrm{MAE} &= \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \\ \mathrm{RMSE} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \\ R^2 &= 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \end{aligned}\right\} \tag{4}$$

where y_i is the true value, $\hat{y}_i$ is the predicted value, and $\bar{y}$ is the mean of the true values. Smaller values of MSE, MAE, and RMSE indicate lower prediction error, whereas an R^2 value closer to 1 indicates better fitting performance.

C. NETWORK PARAMETERS

The main parameter settings of BiLSTM-MTRAN-TCN are listed in Table 3. During training, MSE is used as the loss function, Adam is chosen as the optimizer, and the learning rate is set to 0.00001. The window length is set to 10, meaning that the stock data from the previous 10 trading days are used to predict the closing price of the next trading day.

Table 3. Parameter settings of the BiLSTM-MTRAN-TCN method

Parameter	Value
Batch size	5
Training sequence length	10
BiLSTM hidden size	64
Number of BiLSTM layers	3
Number of Transformer encoder heads	8
Number of Transformer encoder layers	6
Number of neurons in TCN layers	32
TCN convolution stride	1
Number of TCN hidden layers	4
TCN kernel size	7
TCN activation function	Rectified linear unit (ReLU)

V. EXPERIMENTS AND ANALYSIS

This section verifies the effectiveness of the proposed method from four aspects: comparison with representative existing methods, validation of the effectiveness of MTRAN-TCN and BiLSTM, examination of temporal stability, and analysis of generalization ability.

A. EFFECTIVENESS OF MTRAN-TCN AND BILSTM

The stock indices listed in Table 1 are used to verify both the effectiveness of the Transformer modification and the role of introducing BiLSTM. Specifically, BiLSTM-MTRAN-TCN is compared with TRAN, MTRAN-TCN, MTRAN, BiLSTM-MTRAN, BiLSTM, and BiLSTM-TCN. Here, TRAN denotes the original Transformer model, and MTRAN denotes the model composed of the Transformer encoder and a fully connected layer.

The comparison results for the seven methods are shown in Table 4.

Table 4. Comparison of seven methods

Method	MAE	MSE	RMSE	R^2
TRAN	0.284	0.224	0.438	0.778
MTRAN	0.175	0.056	0.229	0.945
MTRAN-TCN	0.173	0.055	0.231	0.945
BiLSTM-MTRAN-TCN	0.087	0.014	0.118	0.986
BiLSTM	0.136	0.035	0.182	0.965
BiLSTM-MTRAN	0.104	0.020	0.138	0.981
BiLSTM-TCN	0.147	0.040	0.195	0.960

As shown in Table 4, TRAN performs markedly worse than the other methods, with an R^2 value of only 0.778, indicating that the original Transformer is not well suited to direct stock price prediction. Compared with TRAN, MTRAN-TCN exhibits substantial improvement: R^2 increases from 0.778 to 0.945, and RMSE decreases from 0.438 to 0.231, demonstrating the effectiveness of the architectural modification.

MTRAN and MTRAN-TCN achieve similar predictive performance, suggesting that the modified Transformer alone does not provide a sufficiently large gain. However, after BiLSTM is introduced before MTRAN-TCN, BiLSTM-MTRAN-TCN performs substantially better than MTRAN-TCN, with RMSE decreasing from 0.231 to 0.118, a reduction of 49.1%, and R^2 increasing from 0.945 to 0.986, an improvement of 4.28%. This result indicates that the introduction of BiLSTM materially improves predictive performance.

From the perspective of hybrid network construction, BiLSTM-MTRAN reduces RMSE by 24.06% and increases R^2 by 1.57% compared with BiLSTM. When BiLSTM is further combined with MTRAN-TCN, RMSE is reduced by 35.22% relative to BiLSTM, and R^2 is improved by 2.09%. These results show that the hybrid network composed of BiLSTM and MTRAN-TCN can achieve better prediction results.

In addition, compared with BiLSTM-MTRAN, BiLSTM-MTRAN-TCN reduces MAE from 0.104 to 0.087, MSE from 0.020 to 0.014, and RMSE from 0.138 to 0.118, while increasing R^2 from 0.981 to 0.986. This further indicates that introducing TCN into the Transformer is beneficial.

In summary, the modification of the Transformer is effective, and introducing BiLSTM before MTRAN-TCN for sequence feature extraction further enhances predictive performance.

B. COMPARISON WITH OTHER METHODS

The BiLSTM-MTRAN-TCN method is further compared with five representative methods: the LSTM method in [10], the BiLSTM method in [35], the CNN-LSTM model in [34], the CNN-BiLSTM-AM model in [8], and the BiLSTM-SA-TCN model in [28]. CNN-BiLSTM-AM is a hybrid model composed of CNN, BiLSTM, and an attention mechanism (AM). BiLSTM-SA-TCN is a hybrid model composed of BiLSTM, self-attention (SA), and TCN.

Comparative experiments are conducted on both stock indices and individual A-share stocks. Each stock is tested five times, and the average of the five runs is reported. All data are standardized before training according to (3), so the reported results are standardized values.

1) A-share Stocks

To further validate the effectiveness of BiLSTM-MTRAN-TCN, it is compared with BiLSTM, CNN-LSTM, CNN-BiLSTM-AM, and BiLSTM-SA-TCN using 14 A-share stocks selected from seven industry categories, as listed in Table 2. The results are shown in Tables 5-11.

Table 5. Comparison of evaluation metrics for China Merchants Bank and Sinolink Securities

China Merchants Bank (SH600036)

Method	MAE	MSE	RMSE	R^2
BiLSTM	0.081	0.012	0.109	0.988
CNN-LSTM	0.265	0.134	0.366	0.861
CNN-BiLSTM-AM	0.277	0.143	0.378	0.851
BiLSTM-SA-TCN	0.110	0.020	0.142	0.979
BiLSTM-MTRAN-TCN	0.042	0.004	0.064	0.996

Sinolink Securities (SH600109)

Method	MAE	MSE	RMSE	R^2
BiLSTM	0.081	0.012	0.109	0.988

Method	MAE	MSE	RMSE	R^2
CNN-LSTM	0.264	0.132	0.364	0.862
CNN-BiLSTM-AM	0.240	0.121	0.348	0.874
BiLSTM-SA-TCN	0.116	0.021	0.146	0.978
BiLSTM-MTRAN-TCN	0.062	0.007	0.083	0.993

Table 6. Comparison of evaluation metrics for Poly Developments and Tibet Urban Development

Poly Developments (SH600048)

Method	MAE	MSE	RMSE	R^2
BiLSTM	0.081	0.012	0.108	0.988
CNN-LSTM	0.265	0.134	0.366	0.860
CNN-BiLSTM-AM	0.293	0.149	0.387	0.845
BiLSTM-SA-TCN	0.119	0.022	0.148	0.977
BiLSTM-MTRAN-TCN	0.056	0.009	0.096	0.990

Tibet Urban Development (SH600773)

Method	MAE	MSE	RMSE	R^2
BiLSTM	0.080	0.011	0.107	0.988
CNN-LSTM	0.257	0.129	0.359	0.866
CNN-BiLSTM-AM	0.241	0.123	0.351	0.872
BiLSTM-SA-TCN	0.107	0.018	0.136	0.981
BiLSTM-MTRAN-TCN	0.096	0.017	0.131	0.982

Table 7. Comparison of evaluation metrics for China Shenhua and Power Investment Energy

China Shenhua (SH601088)

Method	MAE	MSE	RMSE	R^2
BiLSTM	0.084	0.012	0.111	0.987
CNN-LSTM	0.260	0.132	0.363	0.863
CNN-BiLSTM-AM	0.243	0.122	0.349	0.873
BiLSTM-SA-TCN	0.110	0.020	0.140	0.980
BiLSTM-MTRAN-TCN	0.093	0.013	0.112	0.987

Power Investment Energy (SZ002128)

Method	MAE	MSE	RMSE	R^2
BiLSTM	0.083	0.012	0.111	0.987
CNN-LSTM	0.264	0.133	0.365	0.861
CNN-BiLSTM-AM	0.272	0.138	0.372	0.856
BiLSTM-SA-TCN	0.115	0.021	0.145	0.978
BiLSTM-MTRAN-TCN	0.079	0.010	0.098	0.990

Table 8. Comparison of evaluation metrics for CITIC Special Steel and Fangda Carbon

CITIC Special Steel (SZ000708)

Method	MAE	MSE	RMSE	R^2
BiLSTM	0.082	0.012	0.109	0.988
CNN-LSTM	0.256	0.129	0.359	0.866
CNN-BiLSTM-AM	0.255	0.131	0.362	0.864
BiLSTM-SA-TCN	0.109	0.019	0.137	0.981
BiLSTM-MTRAN-TCN	0.075	0.010	0.099	0.990

Fangda Carbon (SH600516)

Method	MAE	MSE	RMSE	R^2
BiLSTM	0.082	0.012	0.109	0.988
CNN-LSTM	0.263	0.133	0.365	0.862
CNN-BiLSTM-AM	0.265	0.135	0.367	0.860
BiLSTM-SA-TCN	0.122	0.023	0.153	0.976
BiLSTM-MTRAN-TCN	0.104	0.016	0.128	0.983

Table 9. Comparison of evaluation metrics for Tianqi Lithium and Yunnan Copper

Tianqi Lithium (SZ002466)

Method	MAE	MSE	RMSE	R^2
BiLSTM	0.081	0.012	0.108	0.988
CNN-LSTM	0.258	0.131	0.361	0.864
CNN-BiLSTM-AM	0.277	0.141	0.376	0.853
BiLSTM-SA-TCN	0.113	0.020	0.140	0.980
BiLSTM-MTRAN-TCN	0.059	0.007	0.084	0.993

Yunnan Copper (SZ000878)

Method	MAE	MSE	RMSE	R^2
BiLSTM	0.081	0.012	0.109	0.988
CNN-LSTM	0.269	0.137	0.370	0.857
CNN-BiLSTM-AM	0.303	0.157	0.396	0.837
BiLSTM-SA-TCN	0.114	0.020	0.142	0.979
BiLSTM-MTRAN-TCN	0.057	0.008	0.088	0.992

Table 10. Comparison of evaluation metrics for PetroChina and Yueyang Xingchang

PetroChina (SH601857)

Method	MAE	MSE	RMSE	R^2
BiLSTM	0.081	0.012	0.108	0.988
CNN-LSTM	0.261	0.133	0.365	0.861
CNN-BiLSTM-AM	0.284	0.146	0.382	0.849
BiLSTM-SA-TCN	0.117	0.021	0.145	0.978
BiLSTM-MTRAN-TCN	0.059	0.006	0.080	0.993

Yueyang Xingchang (SZ300164)

Method	MAE	MSE	RMSE	R^2
BiLSTM	0.083	0.012	0.110	0.987
CNN-LSTM	0.262	0.132	0.364	0.862
CNN-BiLSTM-AM	0.303	0.157	0.396	0.837
BiLSTM-SA-TCN	0.119	0.022	0.147	0.977
BiLSTM-MTRAN-TCN	0.051	0.006	0.076	0.994

Table 11. Comparison of evaluation metrics for BYD and Dongfeng Automobile

BYD (SZ002594)

Method	MAE	MSE	RMSE	R^2
BiLSTM	0.082	0.012	0.109	0.988
CNN-LSTM	0.271	0.137	0.370	0.857
CNN-BiLSTM-AM	0.258	0.130	0.361	0.865
BiLSTM-SA-TCN	0.111	0.019	0.140	0.980
BiLSTM-MTRAN-TCN	0.068	0.008	0.087	0.992

Dongfeng Automobile (SH600006)

Method	MAE	MSE	RMSE	R^2
BiLSTM	0.081	0.012	0.109	0.988
CNN-LSTM	0.256	0.130	0.360	0.865
CNN-BiLSTM-AM	0.257	0.131	0.362	0.863
BiLSTM-SA-TCN	0.121	0.022	0.149	0.977
BiLSTM-MTRAN-TCN	0.048	0.006	0.074	0.994

Tables 5-11 show that the overall errors of CNN-LSTM and CNN-BiLSTM-AM are higher than those of BiLSTM, BiLSTM-SA-TCN, and BiLSTM-MTRAN-TCN, indicating that the two CNN-based methods yield larger fitting errors on the selected stock datasets. Further comparison shows that the errors of BiLSTM are generally lower than those of CNN-LSTM and CNN-BiLSTM-AM, suggesting that the combination of CNN-based local feature extraction and LSTM learning does not improve stock prediction accuracy under the present experimental settings. Improving forecasting accuracy therefore still requires more effective emphasis on features that are closely associated with stock price variation, and the Transformer's multi-head attention mechanism appears advantageous in this regard.

Although BiLSTM-SA-TCN also incorporates self-attention, its overall performance on the 14 stocks remains inferior to that of BiLSTM-MTRAN-TCN. Based on the average results in Tables 5-11, the average MAE, MSE, RMSE, and R^2 of BiLSTM-MTRAN-TCN are 0.068, 0.009, 0.093, and 0.991, respectively. Relative to BiLSTM-SA-TCN, the average RMSE is reduced by 35.3%, and the average R^2 is improved by 0.012. This suggests that, under the present experimental settings, multi-head attention is more effective than plain self-attention.

Overall, BiLSTM-MTRAN-TCN achieves the highest R^2 on 85.7% of the stock datasets and the lowest RMSE on 78.6% of the datasets. Based on the average results of the 14 stocks, its average RMSE decreases by 74.5%, 74.9%, 14.8%, and 35.3% compared with CNN-LSTM, CNN-BiLSTM-AM, BiLSTM, and BiLSTM-SA-TCN, respectively, while the average R^2 increases by 0.129, 0.134, 0.003, and 0.012, respectively. These results indicate that BiLSTM-MTRAN-TCN characterizes stock price fluctuations more effectively and achieves better overall predictive performance than the comparison methods.

2) Stock Indices

In addition to individual A-share stocks, stock index data are introduced to broaden experimental coverage and further validate the proposed method. The five stock indices listed in Table 1 are used as test datasets. For each index and each method, five repeated experiments are conducted, and the average values across the five indices are reported in Table 12.

Table 12. Average evaluation results of six methods

Method	MAE	MSE	RMSE	R^2
CNN-LSTM	0.268	0.120	0.342	0.877
CNN-BiLSTM-AM	0.259	0.116	0.335	0.882
LSTM	0.138	0.037	0.185	0.962
BiLSTM-SA-TCN	0.139	0.036	0.185	0.963
BiLSTM	0.121	0.028	0.162	0.971
BiLSTM-MTRAN-TCN	0.087	0.014	0.118	0.986

As shown in Table 12, BiLSTM-MTRAN-TCN yields the smallest MAE, MSE, and RMSE and the largest R^2 among the six methods, indicating the best average performance in the stock-index experiments. By contrast, CNN-LSTM and CNN-BiLSTM-AM exhibit relatively weak overall performance.

Compared with the other methods in Table 12, BiLSTM-MTRAN-TCN reduces average RMSE by 27.2% to 65.5% and improves R^2 by 0.015 to 0.109. In addition, compared with LSTM, BiLSTM achieves smaller MAE and RMSE and a larger R^2. Specifically, MAE decreases from 0.138 to 0.121, RMSE decreases from 0.185 to 0.162, and R^2 increases from 0.962 to 0.971, which indicates that BiLSTM is superior to LSTM for stock index forecasting. Further, compared with BiLSTM-SA-TCN, BiLSTM-MTRAN-TCN reduces MAE from 0.139 to 0.087 and RMSE from 0.185 to 0.118 while improving R^2 from 0.963 to 0.986. This further suggests that, under the settings of this study, the multi-head attention mechanism is more effective than plain self-attention.

These results indicate that the proposed method more effectively captures price variation trends in both stock-index forecasting and individual stock forecasting.

C. VALIDATION OF GENERALIZATION ABILITY

Based on the experimental results for the 14 A-share stocks, the generalization ability and predictive accuracy of the methods are further analyzed. As shown in Fig. 5, the R^2 values of all six methods are concentrated between 0.8 and 1.0, and no obvious outliers are observed. The R^2 values of BiLSTM are relatively concentrated around 0.988, with both the median and the mean equal to 0.988. By contrast, the R^2 values of BiLSTM-MTRAN-TCN are closer to 1.0, indicating higher predictive accuracy.

Specifically, the R^2 values of BiLSTM-MTRAN-TCN range from 0.982 to 0.996, with a median of 0.992 and a mean of 0.991. The R^2 values of BiLSTM-SA-TCN range from 0.976 to 0.981, with a median of 0.979 and a mean of 0.979. The R^2 values of CNN-LSTM range from 0.857 to 0.866, with a median of 0.862 and a mean of 0.862. The R^2 values of CNN-BiLSTM-AM range from 0.837 to 0.874, with a median of 0.858 and a mean of 0.857. From the distributional perspective, the BiLSTM-MTRAN-TCN results are relatively concentrated, indicating stable behavior on the selected samples. Moreover, its mean and median are both closer to 1.0, which further supports its stronger predictive accuracy.

D. VALIDATION OF TEMPORAL STABILITY

The five stock indices listed in Table 1 are used to examine whether the model exhibits temporal sensitivity, that is, to evaluate its stability across time. Four different time spans are considered: 2009.1-2020.12, 2010.1-2021.12, 2011.1-2022.12, and 2012.1-2023.5.

As shown in Fig. 6, the error metrics of BiLSTM-MTRAN-TCN vary only slightly across the four time periods. The average RMSE values for the four periods are 0.120, 0.112, 0.160, and 0.137, respectively. The corresponding average MAE values are 0.086, 0.083, 0.114, and 0.098, and the average MSE values are 0.016, 0.013, 0.028, and 0.021. The differences in R^2 are likewise small. Figure 7 further shows that the R^2 values of the stock indices change smoothly across the four periods. In the figure, the labels a, b, c, and d denote different time periods, and the same label refers to the same period. The average R^2 values of the four periods are 0.984, 0.987, 0.973, and 0.978, respectively. Overall, the fluctuations in R^2 across time are limited, indicating stable temporal performance.

In addition, the variance of each evaluation metric across the four periods is calculated for each stock index to reflect the dispersion of the results over time. The results are shown in Table 13. The variance of MAE ranges from 0.0002 to 0.0008, the variance of MSE ranges from 0.0000 to 0.0002, the variance of RMSE ranges from 0.0003 to 0.0016, and the variance of R^2 ranges from 0.0000 to 0.0002. Overall, these variances are all close to zero, indicating that the prediction deviations of the five stock indices remain small across different time spans.

Table 13. Variance of the evaluation metrics for five stock indices across four time periods

Index code	Variance of MAE (S^2)	Variance of MSE (S^2)	Variance of RMSE (S^2)	Variance of R^2 (S^2)
000001	0.0006	0.0001	0.0009	0.0001
000002	0.0008	0.0002	0.0016	0.0002
399001	0.0002	0.0000	0.0003	0.0000
399006	0.0002	0.0000	0.0005	0.0000
399300	0.0002	0.0000	0.0004	0.0000

Overall, the error metrics of different stock indices fluctuate only slightly across different time periods, and the prediction results remain stable. This indicates that, over the four time spans considered in this study, BiLSTM-MTRAN-TCN does not exhibit obvious performance degradation and maintains strong temporal stability.

E. OPTIMAL EPOCH VALUE

The training performance of neural network models generally improves as the number of epochs increases, but once training reaches a certain stage, the performance tends to stabilize and may even decline. To analyze the influence of epoch settings on predictive performance, experiments are conducted on the 14 A-share stocks listed in Table 2, and four epoch settings are compared, as shown in Fig. 8. The closer R^2 is to 1, the better the model performance. As shown in Fig. 8, when epoch = 500, R^2 improves substantially relative to epoch = 200 and epoch = 300. When epoch = 600, however, the R^2 values of most stocks remain nearly the same as those at epoch = 500 and do not continue to improve with additional training. Therefore, under the present experimental settings, epoch = 500 is an appropriate parameter choice.

VI. CONCLUSION

This paper proposes the BiLSTM-MTRAN-TCN method for stock closing-price prediction. The method modifies the original Transformer by removing the Input Embedding module, replacing the original decoder structure with TCN layers and fully connected layers, and retaining only the encoder output as the input to subsequent modules. After positional encoding, the data are first processed by BiLSTM to extract sequential dependency features and are then fed into MTRAN-TCN for further modeling. By integrating multiple model structures, the proposed method combines their respective strengths and partially mitigates the limitations of any single architecture, thereby improving predictive accuracy.

Systematic experiments are conducted to evaluate the role of BiLSTM, the effectiveness of the Transformer modification, model accuracy, generalization ability, and temporal stability. The results show that MTRAN-TCN achieves better performance when combined with BiLSTM, indicating that the introduction of BiLSTM contributes positively to stock forecasting.

Compared with LSTM, BiLSTM, CNN-LSTM, CNN-BiLSTM-AM, and BiLSTM-SA-TCN, BiLSTM-MTRAN-TCN achieves the best overall performance in this study. The experiments involve five representative stock indices and 14 A-share stocks, where the 14 stocks are selected from seven industry categories using the proposed bidirectional stock selection strategy. In the stock-index experiments, the average R^2 of BiLSTM-MTRAN-TCN improves by 0.015 to 0.109 relative to the comparison methods. In the A-share stock experiments, the proposed method achieves the highest R^2 on 85.7% of the datasets and the lowest RMSE on 78.6% of the datasets.

In the experiments spanning four different time periods for five stock indices, the error metrics vary only slightly, and the prediction results remain stable overall. These findings indicate that BiLSTM-MTRAN-TCN performs robustly over the considered time spans and exhibits strong accuracy, generalization ability, and temporal stability.

VII. DISCUSSION

Although the proposed method achieves strong predictive performance, there is still room for improvement on some individual stocks. Future research may proceed along the following directions:

- Further optimize the neural network architecture.
- Integrate multiple data sources for prediction, such as stock prices, index data, and fundamental information.
- Incorporate multi-scale temporal information. At present, only one time-window length is considered. Future work may further include different time scales such as 7-day, 30-day, and 150-day windows.

REFERENCES

[1] Z. Hu, W. Liu, J. Bian, X. Liu, and T.-Y. Liu, "Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction," in Proc. 11th ACM Int. Conf. Web Search Data Mining, Marina Del Rey, CA, USA, Feb. 2018, pp. 261-269.

[2] J. Li, S. Pan, and L. Huang, "A machine learning based method for customer behavior prediction," Tehnicki Vjesnik-Tech. Gazette, vol. 26, no. 6, pp. 1670-1676, 2019.

[3] C. Xiao, W. Xia, and J. Jiang, "Stock price forecast based on combined model of ARIMA-LS-SVM," Neural Comput. Appl., vol. 32, no. 10, pp. 5379-5388, May 2020.

[4] P. Yu and X. Yan, "Stock price prediction based on deep neural networks," Neural Comput. Appl., vol. 32, no. 6, pp. 1609-1628, Mar. 2020.

[5] Q. Zhang, C. Qin, Y. Zhang, F. Bao, C. Zhang, and P. Liu, "Transformer-based attention network for stock movement prediction," Exp. Syst. Appl., vol. 202, Sep. 2022, Art. no. 117239.

[6] T. S. Rao and M. M. Gabr, *An Introduction to Bispectral Analysis and Bilinear Time Series Models*. New York, NY, USA: Springer, 2012.

[7] C. N. Babu and B. E. Reddy, "Prediction of selected Indian stock using a partitioning-interpolation based ARIMA-GARCH model," Appl. Comput. Informat., vol. 11, no. 2, pp. 130-143, Jul. 2015.

[8] W. Lu, J. Li, J. Wang, and L. Qin, "A CNN-BiLSTM-AM method for stock price prediction," Neural Comput. Appl., vol. 33, no. 10, pp. 4741-4753, May 2021.

[9] H. Xu, B. Xu, and K. Xu, "Overview of the application of machine learning in stock forecasting," Comput. Eng. Appl., vol. 56, no. 12, pp. 19-24, 2020.

[10] S. Mehtab, J. Sen, and A. Dutta, "Stock price prediction using machine learning and LSTM-based deep learning models," in *Machine Learning and Metaheuristics Algorithms, and Applications* (Communications in Computer and Information Science), vol. 1366, S. M. Thampi, S. Piramuthu, K. C. Li, S. Berretti, M. Wozniak, and D. Singh, Eds. Berlin, Germany: Springer, 2021, pp. 88-106.

[11] D. Wang, X. Liu, and M. Wang, "A DT-SVM strategy for stock futures prediction with big data," in Proc. IEEE 16th Int. Conf. Comput. Sci. Eng., Sydney, NSW, Australia, Dec. 2013, pp. 1005-1012.

[12] Y. Chen and Y. Hao, "A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction," Exp. Syst. Appl., vol. 80, pp. 340-355, Sep. 2017.

[13] Z. Yan, C. Qin, and G. Song, "Stock price prediction of stochastic forest model based on Pearson feature selection," Comput. Eng. Appl., vol. 57, no. 15, pp. 286-296, 2021.

[14] P. Wang, Y. Lou, and L. Lei, "Research on stock price prediction based on BP wavelet neural network with Mexico hat wavelet basis," in Proc. Int. Conf. Educ., Econ. Manag. Res. (ICEEMR), Singapore, 2017, pp. 99-102.

[15] Y. Hu, "Stock market timing model based on convolutional neural network-A case study of Shanghai composite index," Finance Economy, vol. 26, no. 4, pp. 71-74, 2018.

[16] S. Jain, R. Gupta, and A. A. Moghe, "Stock price prediction on daily stock data using deep neural networks," in Proc. Int. Conf. Adv. Comput. Telecommun. (ICACAT), Bhopal, India, Dec. 2018, pp. 1-13.
[17] Q. Yang and C. Wang, "Research on global stock index prediction based on deep learning LSTM neural network," Stat. Res., vol. 36, no. 3, pp. 65-77, 2019.
[18] J. Qiu, B. Wang, and C. Zhou, "Forecasting stock prices with long-short term memory neural network based on attention mechanism," PLoS ONE, vol. 15, no. 1, Jan. 2020, Art. no. e0227222.
[19] A. Vaswani, "Attention is all you need," in Proc. 31st Conf. Neural Inf. Process. Syst., Long Beach, CA, USA, 2017, pp. 1-11.
[20] Q. Ding, S. Wu, H. Sun, J. Guo, and J. Guo, "Hierarchical multi-scale Gaussian transformer for stock movement prediction," in Proc. 29th Int. Joint Conf. Artif. Intell., Jul. 2020, pp. 4640-4646.
[21] R. M. Farsani and E. Pazouki, "A transformer self-attention model for time series forecasting," J. Electr. Comput. Eng. Innov., vol. 9, no. 1, pp. 1-10, 2021.
[22] Z.-Y. Peng and P.-C. Guo, "A data organization method for LSTM and transformer when predicting Chinese banking stock prices," Discrete Dyn. Nature Soc., vol. 2022, pp. 1-8, Jan. 2022.
[23] C. Wang, Y. Chen, S. Zhang, and Q. Zhang, "Stock market index prediction using deep transformer model," Exp. Syst. Appl., vol. 208, Dec. 2022, Art. no. 118128.
[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735-1780, Nov. 1997.
[25] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks: A unified approach to action segmentation," in *Computer Vision-ECCV* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2016, pp. 47-54.
[26] Z. Chen, J. Liu, and B. Zhao, "Multi factor urban road network taxi demand prediction based on GCN and TCN," Control Decis., vol. 38, no. 4, pp. 1031-1038, 2023.
[27] J. Wang, Z. Gao, and C. Shan, "Multivariate prediction of the Yellow River runoff based on the TCN-attention model," People's Yellow River, vol. 11, pp. 20-25, 2022.
[28] Y. Zhiyong, Y. Yuxi, and Z. Yu, "Application of BiLSTM-SA-TCN time series model in stock forecasting," J. Nanjing Inf. Eng. Univ., pp. 1-12, Apr. 2023.
[29] J. Deng, F. Zhao, and X. Wang, "MTICA-AEO-SVR model for stock price forecasting," Comput. Eng. Appl., vol. 58, no. 8, pp. 257-263, 2022.
[30] X. Li, C. Cui, and G. Song, "Stock trend prediction method based on temporal hypergraph convolutional neural network," J. Comput. Appl., vol. 42, no. 3, pp. 797-803, 2022.
[31] T. Zhang, Z. Tang, and J. Wu, "Forecasting method of stock index based on optimized KELM model," Statist. Decis., vol. 37, no. 13, pp. 148-150, 2021.
[32] Y. Gao, R. Wang, and E. Zhou, "Stock prediction based on optimized LSTM and GRU models," Sci. Program., vol. 2021, pp. 1-8, Sep. 2021.
[33] A. Yadav, C. K. Jha, and A. Sharan, "Optimizing LSTM for time series prediction in Indian stock market," Proc. Comput. Sci., vol. 167, pp. 2091-2100, Jan. 2020.
[34] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, "A CNN-LSTM-based model to forecast stock prices," Complexity, vol. 2020, pp. 1-10, Nov. 2020.
[35] A. Zeng and W. Nie, "Stock recommendation system based on deep bidirectional LSTM," Comput. Sci., vol. 46, no. 10, pp. 84-89, 2019.