WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 8, Number 1, Year 2026

DOI: https://doi.org/10.62051/ijcsit.v8n1.07

Journal homepage: https://wepub.org/index.php/IJCSIT/index

BiLSTM-SDTCN-AutoCorr: A Hybrid Model for Stock Price Prediction Integrating Sequence Decomposition and Autocorrelation Attention

Weijun Fan

School of Science, Jimei University, Xiamen, 361021, China

ryan@jmu.edu.cn

ABSTRACT

Stock price prediction faces substantial challenges due to nonlinearity, non-stationarity, and noise contamination. Traditional econometric models and early Deep Learning methods struggle to effectively capture complex temporal patterns. This paper proposes a novel hybrid Neural Network, BiLSTM-SDTCN-AutoCorr, which refines a BiLSTM–Transformer backbone: a sequence decomposition module partitions the input series into trend and seasonal components to filter noise and enhance pattern separation; the vanilla self-attention mechanism is replaced by autocorrelation attention to efficiently capture periodic dependencies via the Fast Fourier transform; and the Transformer decoder is modified into Temporal Convolutional Network layers to strengthen local sequence modeling. The model is evaluated on five stock index datasets, and the results demonstrate significant superiority across evaluation metrics. The proposed model offers an efficient and robust solution for stock prediction with potential practical applicability.

KEYWORDS

Transformer; Hybrid Neural Network; Stock Price Prediction; Auto-Correlation Attention; Deep Learning

1. INTRODUCTION

As a core component of the modern financial system, the stock market's price fluctuations not only provide a composite reflection of macroeconomic conditions, market supply–demand dynamics, and investor sentiment, but also directly affect the optimization of capital allocation efficiency and the investment returns of market participants [1]. Therefore, accurate stock price prediction has long been a shared focus of research in both academia and the financial industry [2, 3]. However, stock price time series are not simple linear time series data; rather, they exhibit pronounced nonlinearity and non-stationarity, and factors such as policy shocks and market turbulence introduce strong noise interference, rendering the forecasting process highly challenging [4, 5].

Early traditional econometric models, such as ARIMA and GARCH, although demonstrating certain advantages in short-term forecasting, often struggle to capture the non-periodic features and complex nonlinear relationships in stock data [6]. Moreover, they fail to adequately reflect the true distribution of stock data, resulting in insufficient performance in long-term prediction and volatility modeling.

In recent years, the rise of deep learning methods has provided a new paradigm for time series forecasting, enabling the automatic learning of complex nonlinear models from raw data without the need for extensive feature engineering [7]. Classic models such as CNN, RNN, LSTM, and BiLSTM excel in capturing sequence dependencies [8-10]. In 2020, Liu and Long et al. used a hybrid EWTdpLSTM-PSO-ORELM framework to predict the stock closing prices of S&P500, CMSB, and DJIA, demonstrating that this framework exhibits excellent prediction accuracy, significantly outperforming baseline models such as BP and single LSTM [11]. Similarly, in 2023, Gülmez et al. proposed a deep LSTM model optimized by the Artificial Rabbits Optimization algorithm for predicting DJIA index stock prices. The experimental results showed that this model significantly outperforms baseline models such as ANN and LSTM-GA in evaluation metrics including MSE, MAE, MAPE, and R^2 [12]. Due to their stronger feature extraction and nonlinear modeling capabilities, deep learning models can mine potential patterns from vast and complex historical data, effectively capturing price change trends and significantly improving prediction accuracy, making them one of the important research directions in stock price prediction [13].

Nowadays, the attention mechanism has become the mainstream approach for addressing time series forecasting in financial markets. Inspired by advancements in NLP and Computer Vision, scholars have begun exploring the potential of the Transformer architecture in time series modeling [14, 15]. In 2022, Zhang et al. introduced TEANet, an attention network based on the Transformer encoder, which addresses financial time series dependency issues using small-sample data over a 5-day window. By fusing textual data from the X platform with stock price data, the model achieved superior accuracy in stock movement prediction across four datasets compared to baseline models such as ARIMA and CapTE. Furthermore, trading simulations demonstrated its ability to significantly enhance returns, indicating practical application value [16]. Yang et al. (2025) proposed an Adaptive Sharpe Ratio Optimized Time Fusion Transformer (TFT-ASRO) model, which integrates multi-sensor real-time market data and financial indicators to enable multi-task learning for stock Sharpe ratio prediction. The model improved accuracy by 18% over existing deep learning baselines across different time spans, performing particularly well in volatile markets [17].

We can observe that scholars have proposed numerous Transformer-based models for stock prediction. However, the Transformer attention mechanism in traditional models primarily relies on dot-product attention, with a computational complexity of $O(L^2)$, which exhibits low efficiency in capturing the periodicity and long-term dependencies of time-series data [18, 19]. This is particularly evident in financial time series, where the data often contain seasonal and trend noise, making it difficult to disentangle the entangled patterns of trends and fluctuations in stock time series, thereby leading to insufficient prediction stability [20]. Additionally, when adapting the traditional Transformer decoder for financial time-series prediction, structural redundancy exists, and it fails to fully integrate the sequence dependency enhancement capabilities of TCN, resulting in challenges in balancing long-period prediction accuracy and efficiency.

Hence, to mitigate the shortcomings of current stock price forecasting frameworks in handling long-sequence dependencies, complex time-series pattern decomposition, and information utilization efficiency, This study utilizes BiLSTM-Transformer as the core framework and implements specific enhancements, introducing an innovative enhanced hybrid neural network model—BiLSTM-SDTCN-AutoCorr—to boost the precision and resilience of stock price forecasting.

To encapsulate, the primary contributions of this study include the following:

Innovative architecture optimization: By incorporating an auto-correlation attention and a sequence decomposition module, which are respectively used to model long-term periodic dependencies and to separate trend and seasonal features, thereby adapting to the non-linear and non-stationary characteristics of stock time series.

Integration of advantages from multiple models: The proposed model ingeniously combines the sequence memory capabilities of BiLSTM, the auto-correlation attention of Transformer, and the local convolutional modeling advantages of Temporal Convolutional Network (TCN), forming an efficient hybrid architecture.

Empirical validation and performance improvement: On five Chinese stock index datasets (including the Shanghai Composite Index, CSI 300, etc.), comparative experiments with baseline models (BiLSTM, Transformer, BiLSTM+CNN, and BiLSTM+Transformer) demonstrate the proposed model's significant superiority in metrics such as MSE, MAE, RMSE, and R^2.

The structure of the paper is organized as follows: Section 2 reviews related algorithms; Section 3 details the proposed model architecture and related methods such as data preprocessing; Section 4 introduces the experimental setup and datasets; Section 5 presents the results analysis and discussion; Section 6 concludes the paper.

2. RELATED WORK

2.1. Series Decomposition Module and Auto-Correlation Attention Mechanism

The Autoformer model, proposed by Wu et al. in 2021, provides key insights for addressing the core challenges in long-term time series forecasting [20]. This model breaks through the architectural design of the traditional Transformer by innovatively introducing the Series Decomposition Block, which dynamically decomposes the time series data X into a trend component X_t (reflecting long−term stable changes) and a seasonal component X_s (reflecting short−term periodic fluctuations). The decomposition process is implemented using a Moving Average filter, as shown in Equation (1).

$$\begin{gathered} X_t = \mathrm{AvgPool}\left(Padding(X)\right) \\ X_s = X - X_t \end{gathered} \tag{1}$$

This decomposition enables the model to separately process the long-term trends and short-term patterns in the sequence, particularly in non-stationary financial data, which can mitigate noise interference and enhance prediction stability, avoiding the shortcomings of pre-decomposition methods that overlook interactions among future components.

Autoformer further introduces the auto-correlation mechanism as an alternative to traditional self-attention. This mechanism is based on the autocorrelation function of the sequence, computing correlations between similar subsequences rather than dot-product similarity, with the specific calculation as shown in Equation (2).

$$R_{XX}(\tau) = \lim_{L\to\infty} \frac{1}{L}\sum_{t=1}^{L} X_t X_{t-\tau} \tag{2}$$

Where, $R_{XX}(\tau)$ reflects the similarity between X_t and its lagged sequence $X_{t-\tau}$ by τ periods. Specifically, the autocorrelation attention mechanism employs FFT to efficiently compute correlations, and selects the most relevant key−value pairs through Time Delay Aggregation and Stochastic Selection, with the specific computation as shown in Equation (3).

$$\begin{gathered} \tau_1, \cdots, \tau_k = \mathrm{argTopk}\left(R_{Q,K}(\tau)\right) \\ \widehat{R}_{Q,K}(\tau_1), \cdots, \widehat{R}_{Q,K}(\tau_k) = \mathrm{SoftMax}\left(R_{Q,K}(\tau_1), \cdots, R_{Q,K}(\tau_k)\right) \\ \mathrm{Auto-Correlation}(Q,K,V) = \sum_{i=1}^{k} \mathrm{Roll}(V,\tau_i)\widehat{R}_{Q,K}(\tau_i) \end{gathered} \tag{3}$$

Where $\mathrm{argTopk}$ is to get the arguments of the Topk autocorrelations and let $k = \lfloor c \times \log L \rfloor$, c is a hyper-parameter. $R_{Q,K}$ is autocorrelation between series Q and K. $\mathrm{Roll}(\mathrm{X},\tau)$ represents the operation to X with time delay τ, during which elements that are shifted beyond the first position are re-introduced at the last position. This design reduces the computational complexity to $O(n\log n)$, and naturally captures the periodic dependencies in the sequence.

3. METHODOLOGY

In this section, we first introduce the data preprocessing methods and elaborate on the improvement strategies for the BiLSTM-Transformer structure. Subsequently, we describe the overall architecture of the proposed neural network. The detailed description of the specific datasets will be presented in the experimental section.

3.1. Data Preprocessing

Within the domain of financial time series examination, proficient data preparation plays a vital role in developing dependable and precise forecasting models. This subsection outlines a series of preprocessing steps applied to the stock datasets. These steps primarily include data normalization, feature engineering (adding new feature indicators), and preparing the datasets for time series modeling.

3.1.1. Data Normalization

Due to the substantial differences in the scales of features in the raw stock datasets, we apply Z-score standardization to the stock data to ensure that all features contribute equally to the learning process and to enhance model convergence. The specific calculation is shown in Equation (4).

$$y_i = \frac{x_i - \bar{x}}{s} \tag{4}$$

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

3.1.2. Feature Engineering

To enhance the model's predictive performance, additional technical indicators are generated based on stock price data: the difference between the current day's closing price and the previous day's closing price is used as the "price change amount," reflecting absolute price changes; the percentage change relative to the previous day's closing price is used as the "price change rate," reflecting relative price changes; meanwhile, moving averages are calculated to capture short-term trends in the stock price series. Stock prices are averaged over different moving windows, with the specific calculation formula shown in Equation (5).

$$\mathrm{MA}_t^w = \frac{1}{w}\sum_{i=0}^{w-1} x_{t-i} \tag{5}$$

Where MA_t^w represents the moving average at time step t, and w is the window size. In our experiments, we computed the 5-day and 10-day window moving averages to capture short-term trends in the stock price sequences.

3.1.3. Data Preparation

In preparing data for time series forecasting, we reshape the raw dataset into overlapping sequences of fixed length. Specifically, given a sequence length L, the input sequence at time t includes stock prices and engineered features from $t-L$ to $t-1$, while the target output is the stock price at time t. Formally, the input sequence X_t and the corresponding target y_t can be defined as:

$$Xt = \{(p_{t-L}, f_{t-L}), \dots, (p_{t-1}, f_{t-1})\}, yt = pt \tag{6}$$

Where p denotes the stock price, and f represents the engineered features. Multiple input-output pairs are generated from historical stock price data using the sliding window method. In this experiment, we set the sequence length L = 20, meaning that the input consists of the stock prices from the past 20 days to predict the stock price on the 21st day.

3.2. The BiLSTM-SDTCN-AutoCorr Method in This Paper

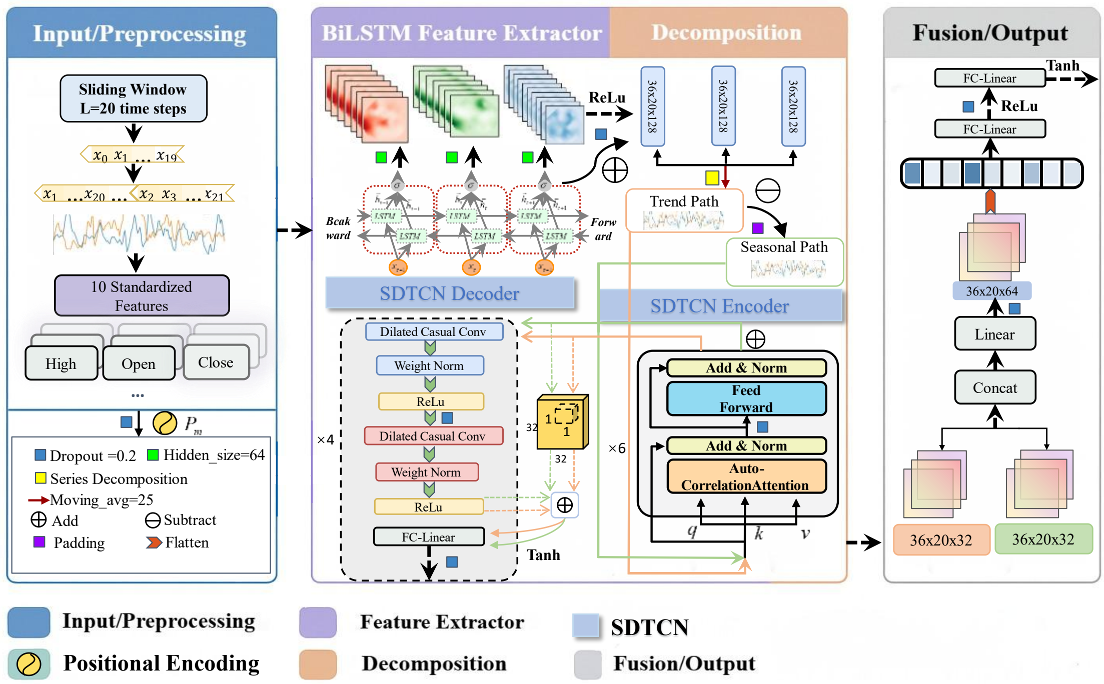

Figure 1. The Structure of BiLSTM-SDTCN-AutoCorr

This study proposes a hybrid neural network architecture for stock price prediction, which integrates Bidirectional Long Short-Term Memory (BiLSTM), an improved traditional Transformer structure, and introduces a sequence decomposition strategy to implement a dual-path encoder-decoder design, ultimately incorporating an autocorrelation attention mechanism to achieve fine-grained modeling of time series data. The specific structure is illustrated in Figure 1.

3.2.1. Input Layer and Positional Encoding

The input data is first injected with temporal order information through sine-cosine positional encoding. This positional encoding module is placed before the BiLSTM layers to enhance the model's perception of sequence dependencies and avoid interference in later encoding stages.

3.2.2. BiLSTM Feature Extraction

The positionally encoded data is fed into a 3-layer bidirectional LSTM for preliminary temporal feature extraction. Each layer contains 64 hidden units, producing 128-dimensional features after bidirectional processing, with residual connections, layer normalization, and Dropout regularization employed to stabilize training and prevent overfitting.

3.2.3. Sequence Decomposition Module

Following the BiLSTM output, a sequence decomposition module is introduced to decompose the feature sequence into trend and seasonal components. This decomposition not only reduces the complexity of the sequence but also allows the model to perform independent modeling at different time scales, thereby improving the robustness of predictions.

3.2.4. Adjustments to the encoder and decoder

Removal of the input embedding module: The input embedding module in traditional Transformers is primarily used for text vectorization (e.g., word embeddings). However, in stock price prediction, the input consists of numerical time series, which do not require text vectorization. Therefore, we removed this module and directly fed the raw features into the positional encoding layer. This modification simplifies the architecture, reduces the number of parameters, and avoids unnecessary conversion noise, thereby enhancing the model's sensitivity to numerical data.

Modification of the Transformer model's decoder: The decoder in traditional Transformers relies on multi-layer attention mechanisms and masked inputs, making it suitable for generative tasks (e.g., sequence-to-sequence translation). However, stock prediction does not require generative decoding. Thus, we replaced the decoder with Temporal Convolutional Network (TCN) layers, fully connected layers, and a Tanh activation function. This replacement enables the decoding process to emphasize the capture of temporal causal relationships, while the Tanh function maps the output to the [-1, 1] range, accommodating the normalization needs of price fluctuations.

Integration of the Auto-Correlation Mechanism: Traditional multi-head attention is based on dot-product similarity computation, whereas Auto-Correlation relies on the FFT to efficiently compute sequence correlations, and discovers periodic dependencies (such as seasonal fluctuations in the stock market) through Time Delay Aggregation, as well as aggregates the most important delay patterns using a Stochastic Selection strategy.

Enhanced Residual Connection Architecture: To address the gradient vanishing problem in deep Transformers, multi-level residual connections are introduced. Each encoder layer incorporates three sub-layers: the Auto-Correlation layer, the feedforward network layer, and the deep residual connection layer, along with a global residual connection to improve gradient flow in deep networks.

3.2.5. Temporal Convolutional Network Decoder

The causal convolution in the TCN layer ensures that the model relies only on past data when predicting the current time step, avoiding leakage of future information [21]. The specific computation is shown in Equation (7).

$$F(s) = \sum_{i=0}^{k-1} f(i) x_{s-di} \tag{7}$$

Meanwhile, the dilated convolutions in TCN can introduce a dilation factor into the convolution kernel to expand the receptive field, thereby efficiently capturing long-range dependencies without the need to stack excessive layers. Mathematically, the output of the dilated convolution is shown in Equation (8).

$$y_t = \sum_{k=0}^{K-1} w_k \cdot x_{t-d\cdot k} \tag{8}$$

Where, y_t is the output at time step t, w_k is the convolutional weight, d is the dilation factor, and K is the kernel size. As the number of layers increases, the dilation factor typically grows exponentially, causing the receptive field to expand exponentially. This enables Temporal Convolutional Network to capture long-term dependencies without increasing the network depth.

4. EXPERIMENTAL SETUP

This section commences with an outline of the experimental configuration, encompassing the datasets employed, metrics for performance assessment, and key model parameters.

4.1. Dataset Description

To ensure the scalability of the model, this paper employs five datasets for experiments. The selected index stock datasets are shown in Table 1.

Table 1. The selected index stock datasets

NUMBER	Index Name	Index Code
1	Shanghai Composite Index	000001.XSHG
2	Shenzhen Component Index	399001.XSHE
3	CSI 300	000300.XSHG
4	CSI 800	000906.XSHG
5	ChiNext Index	399006.XSHE

The index stock data is sourced from AKshare, an open-source financial data interface library in Python. The data for the five index stocks used in the experiments covers the time period from February 1, 2012, to February 28, 2025. We utilize historical trading data as the dataset, including closing price, highest price, lowest price, opening price, change amount, change percentage, trading volume, trading amount, as well as two related technical indicators (5-day moving average and 10-day moving average). Each stock's data consists of 3178 samples.

4.2. Performance Evaluation

In this study, we employ four statistical evaluation metrics to compare the performance of the relevant models, namely MSE, MAE, RMSE, and R^2.

5. NETWORK PARAMETERS

Table 2. The parameter settings for the BiLSTM-SDTCN-AutoCorr method

Parameter	Value
Input features	10
Hidden size of BiLSTM	64
Number of BiLSTM layers	3
Number of transformer encoder heads	8
Number of transformer encoder layers	6
Transformer feed-forward dim	512
TCN kernel size	7
Decomposition kernel size	25
Auto-correlation factor	1
Dropout	0.2
Epochs	400
Batch size	36
Learning rate	0.0001
Optimizer	adam
Loss function	mse
Window size	20

The parameter settings for the BiLSTM-SDTCN-AutoCorr method in this experiment are shown in Table 2. The number of input features is 10, the number of units in the BiLSTM layers is 64, with a total of 3 layers, the loss function is MSE, the optimizer is Adam, and the learning rate is 0.0001. The window size is 20, meaning the closing price of the stock for the next day is predicted based on the stock data from the previous 20 days, and the batch size is 36.

6. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we present comprehensive experiments and analyses conducted to evaluate the performance of the proposed BiLSTM-SDTCN-AutoCorr method framework. To investigate the predictive performance of this model, we selected four benchmark models for comparison, including BiLSTM, Transformer, BiLSTM+CNN, and BiLSTM+Transformer. We conducted comparative experiments between BiLSTM-SDTCN-AutoCorr and these models on four different index stock datasets to verify its superiority in the stock price prediction task.

6.1. Overall Performance Comparison

Table 3. Prediction Metrics of Five Methods on Shanghai Composite Index

Model	MSE	MAE	RMSE	R^2
BiLSTM	0.0037	0.0427	0.0609	0.941
BiLSTM+CNN	0.0106	0.0732	0.1038	0.873
Transformer	0.0068	0.0667	0.0826	0.918
BiLSTM+Transformer	0.0054	0.0596	0.0732	0.931
BiLSTM-SDTCN-AutoCorr	0.0021	0.0312	0.0458	0.964

Table 4. Prediction Metrics of Five Methods for Four Other Stock Indices

Stock	Model	MSE	MAE	RMSE	R^2
000300.XSHG	BiLSTM-SDTCN-AutoCorr	0.00116	0.0267	0.0340	0.968
000300.XSHG	BiLSTM	0.00342	0.0398	0.0585	0.941
000300.XSHG	BiLSTM+CNN	0.00679	0.0612	0.0824	0.913
000300.XSHG	Transformer	0.00623	0.0634	0.0790	0.925
000300.XSHG	BiLSTM+Transformer	0.00499	0.0567	0.0706	0.939
000906.XSHG	BiLSTM-SDTCN-AutoCorr	0.00109	0.0254	0.0330	0.971
000906.XSHG	BiLSTM	0.00320	0.0376	0.0565	0.963
000906.XSHG	BiLSTM+CNN	0.00623	0.0587	0.0790	0.928
000906.XSHG	Transformer	0.00579	0.0601	0.0761	0.931
000906.XSHG	BiLSTM+Transformer	0.00457	0.0543	0.0676	0.944
399006.XSHE	BiLSTM-SDTCN-AutoCorr	0.00142	0.0312	0.0377	0.958
399006.XSHE	BiLSTM	0.00416	0.0456	0.0645	0.946
399006.XSHE	BiLSTM+CNN	0.00786	0.0689	0.0886	0.907
399006.XSHE	Transformer	0.00723	0.0712	0.0851	0.912
399006.XSHE	BiLSTM+Transformer	0.00579	0.0623	0.0761	0.928
399001.XSHE	BiLSTM-SDTCN-AutoCorr	0.00133	0.0298	0.0365	0.962
399001.XSHE	BiLSTM	0.00368	0.0412	0.0606	0.958
399001.XSHE	BiLSTM+CNN	0.00723	0.0654	0.0851	0.916
399001.XSHE	Transformer	0.00679	0.0678	0.0824	0.919
399001.XSHE	BiLSTM+Transformer	0.00523	0.0589	0.0723	0.939

All models performed predictions based on the data of five index stocks as shown in Table 1. For each stock, five independent tests were executed, and the average of the five results was calculated as the final prediction output. Prior to training, all data were standardized using Equation (5). Table 3 presents the prediction evaluation metrics of the five methods on the Shanghai Composite Index (stock code: 000001.XSHG) dataset; Table 4 shows the performance of the aforementioned five methods on the other four index stocks.

The experimental results demonstrate that the proposed BiLSTM-SDTCN-AutoCorr model significantly outperforms all baseline models on the test set. The predictive capabilities of the aforementioned five methods, ranked in descending order, are as follows: BiLSTM-SDTCN-AutoCorr, BiLSTM, BiLSTM+Transformer, Transformer, and BiLSTM+CNN.

As shown in Table 3, BiLSTM-SDTCN-AutoCorr achieves the best values across key metrics including MSE, MAE, RMSE, and R^2. Specifically, compared to the best-performing baseline model BiLSTM, BiLSTM-SDTCN-AutoCorr reduces MSE by 43.2% (from 0.0037 to 0.0021), MAE by 26.9% (from 0.0427 to 0.0312), and RMSE by 24.8% (from 0.0609 to 0.0458), while increasing R^2 by 2.3% (from 0.942 to 0.964). These improvements not only reflect enhanced prediction accuracy but also highlight the model's stronger explanatory power for data variability. This overall performance advantage stems from the innovative architectural design of BiLSTM-SDTCN-AutoCorr, which more effectively captures the complex dynamic patterns in time series, thereby achieving higher accuracy and robustness in stock price prediction tasks. Figure 2 and Figure 3 more intuitively illustrate the significant advantages of the BiLSTM-SDTCN-AutoCorr model in terms of fitting quality and prediction precision, particularly achieving a qualitative improvement in error control. The BiLSTM-SDTCN-AutoCorr model excels in capturing the overall upward and downward trends of the Shanghai Composite Index, with the predicted curve closely aligning with the actual value curve, demonstrating the model's effective modeling of long-term dependencies. In the time step interval of 0-50, the model accurately predicts the downward trend of stock prices from 3100 points to 2900 points; in the time step interval of 500-550, it captures the rapid upward trend from 2700 points to 3400 points; and in the time step interval of 125-240, it precisely reproduces the oscillatory consolidation in the 3100-3400 point range. This trend-capturing capability benefits from the model's effective handling of long-term dependencies.

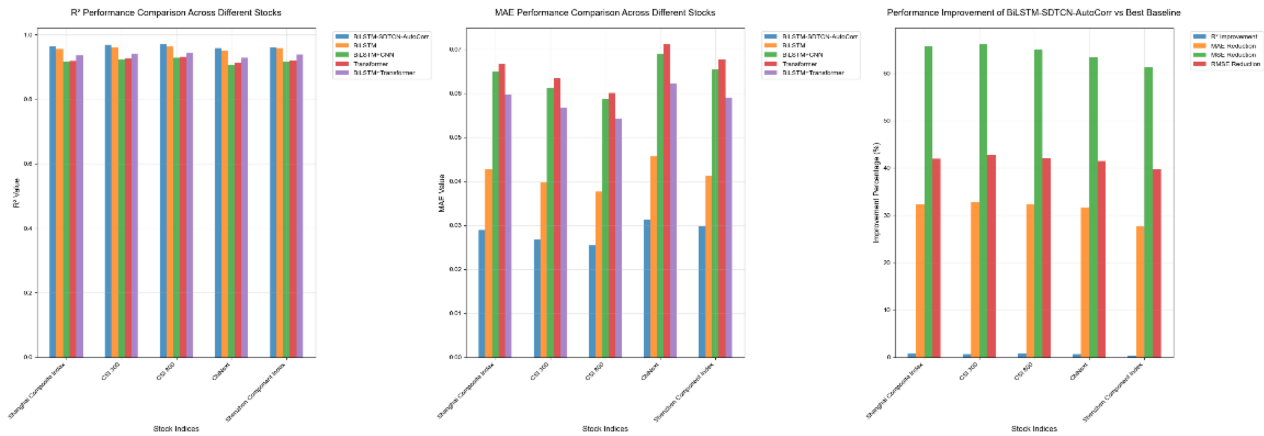

Figure 2. Comparison of Performance Improvement for Prediction Models

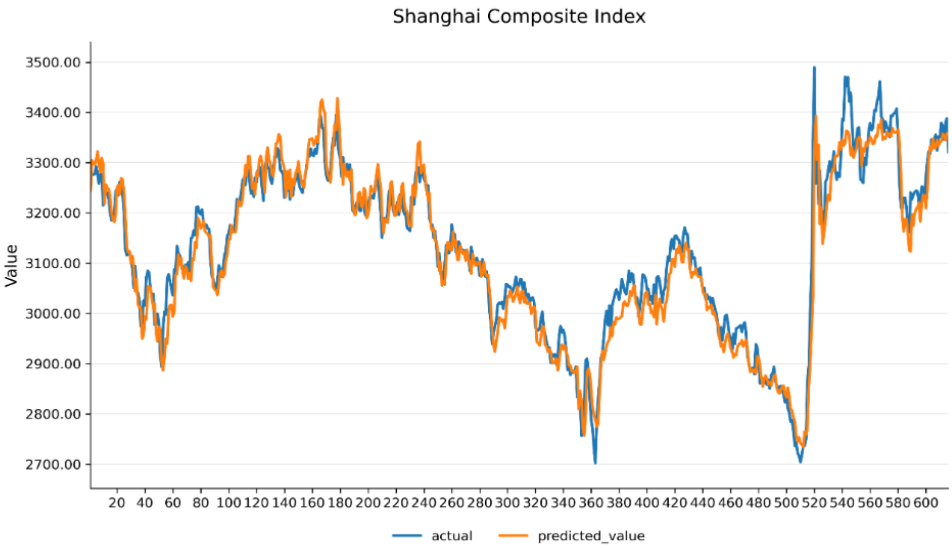

Figure 4. Model Prediction Fitting Curve Chart

6.2. Ablation Experiments

To quantify the role of each component in model performance, we conducted ablation experiments on the CSI 300 (stock code: 000300.XSHG) dataset, with results detailed in Table 5. The complete BiLSTM-SDTCN-AutoCorr model achieved MSE, MAE, and R^2 values of 0.00113, 0.0340, and 0.969, respectively. Removing the Series Decomposition component increased MSE to 0.00168 and decreased R^2 to 0.952, resulting in an overall performance degradation of approximately 1.3%, which confirms the core contribution of this component in enhancing prediction accuracy. Replacing Auto-Correlation with a traditional attention mechanism led to a further performance degradation of 1.67%, thereby highlighting its unique advantage in capturing periodic dependencies in time series. Removing TCN as the decoding layer reduced R^2 to 0.935 (a performance decline of 3.03%), verifying the effectiveness of improving the Transformer decoder to a TCN layer. These results indicate that the TCN decoding layer contributes the most, followed by the auto-correlation mechanism and the series decomposition design, which collectively form the foundation of the model's performance.

Table 5. Performance Metrics of BiLSTM-SDTCN-AutoCorr Model Variants

Model Variant	R^2	MAE	MSE	RMSE	Performance Drop (%)
Completed BiLSTM-SDTCN-AutoCorr	0.969	0.0340	0.00113	0.0263	0.0
Without Series Decomposition	0.952	0.0356	0.00168	0.041	1.3
Without Auto-Correlation	0.948	0.0389	0.00192	0.0438	1.67
Without TCN Decoder	0.935	0.0423	0.00216	0.0464	3.03

7. CONCLUSION

In this study, we introduce an innovative mixed neural network architecture, BiLSTM-SDTCN-AutoCorr, which integrates BiLSTM for sequence feature extraction, a sequence decomposition module for trend-seasonal separation, auto-correlation attention for modeling periodic dependencies, and a TCN decoder for local convolution enhancement, thereby systematically optimizing the traditional BiLSTM-Transformer framework. The core innovations of this model lie in its dual-path encoder-decoder architecture and the integration of the Auto-Correlation mechanism: the former enables multi-scale feature separation and fusion, while the latter efficiently captures long-term periodic patterns using Fourier transforms, significantly reducing computational complexity and enhancing prediction robustness.

Experimental results on five Chinese stock index datasets (including the Shanghai Composite Index and CSI 300) demonstrate the model's superiority: compared to baseline models (such as BiLSTM, Transformer, BiLSTM+CNN, and BiLSTM+Transformer), BiLSTM-SDTCN-AutoCorr achieves average improvements of 20%-40% across MSE, MAE, RMSE, and R^2 metrics. These results not only highlight the model's advantages in capturing complex dynamics in stock markets (such as trend fluctuations and short-term oscillations) but also provide a solid foundation for its applications in real-world financial decision-making (such as risk assessment and investment strategy optimization).

Although the model exhibits remarkable generalization ability and prediction accuracy, its sensitivity to external factors (such as sudden events or macroeconomic policies) remains to be explored. Future research could extend to multimodal data fusion (such as integrating news sentiment analysis or macroeconomic indicators) or incorporate federated learning to enhance cross-market prediction performance under privacy protection.

REFERENCES

[1] Javaid, H.A. (2024) Ai-driven predictive analytics in finance: Transforming risk assessment and decision-making. Advances in Computer Sciences. 7 (1), 1–9.

[2] Thakkar, A. and Chaudhari, K. (2021) Fusion in stock market prediction: A decade survey on the necessity, recent developments, and potential future directions. Information Fusion. 65 95–107.

[3] Hu, Z., Liu, W., Bian, J., Liu, X., & Liu, T. Y. (2018, February). Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In Proceedings of the eleventh ACM international conference on web search and data mining (pp. 261-269).

[4] Bontempi, G., Ben Taieb, S., and Le Borgne, Y.-A. (2013) Machine Learning Strategies for Time Series Forecasting. in: M.-A. Aufaure, E. Zimányi (Eds.), Bus. Intell. Second Eur. Summer Sch. EBISS 2012 Bruss. Belg. July 15-21 2012 Tutor. Lect., Springer Berlin Heidelberg, Berlin, Heidelbergpp. 62–77.

[5] Stărică, C. and Granger, C. (2005) Nonstationarities in Stock Returns. The Review of Economics and Statistics. 87 (3), 503–522.

[6] Babu, C.N. and Reddy, B.E. (2015) Prediction of selected Indian stock using a partitioning–interpolation based ARIMA–GARCH model. Applied Computing and Informatics. 11 (2), 130–143.

[7] Thakkar, A. and Chaudhari, K. (2021) A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. Expert Systems with Applications. 177 114800.

[8] Zhang, J., Ye, L., and Lai, Y. (2023) Stock price prediction using CNN-BiLSTM-Attention model. Mathematics. 11 (9), 1985.

[9] Lu, W., Li, J., Wang, J., and Qin, L. (2021) A CNN-BiLSTM-AM method for stock price prediction. Neural Computing and Applications. 33 (10), 4741–4753.

[10] Kamalov, F. (2020) Forecasting significant stock price changes using neural networks. Neural Computing and Applications. 32 (23), 17655–17667.

[11] Liu, H. and Long, Z. (2020) An improved deep learning model for predicting stock market price time series. Digital Signal Processing. 102 102741.

[12] Gülmez, B. (2023) Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. Expert Systems with Applications. 227 120346.

[13] Jiang, W. (2021) Applications of deep learning in stock market prediction: recent progress. Expert Systems with Applications. 184 115537.

[14] Liu, J., Lin, H., Liu, X., Xu, B., Ren, Y., Diao, Y., & Yang, L. (2019, August). Transformer-based capsule network for stock movement prediction. In Proceedings of the first workshop on financial technology and natural language processing (pp. 66-73).

[15] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020) An image is worth 16x16 words: Transformers for image recognition at scale. arXiv Preprint arXiv:2010.11929.

[16] Zhang, Q., Qin, C., Zhang, Y., Bao, F., Zhang, C., and Liu, P. (2022) Transformer-based attention network for stock movement prediction. Expert Systems with Applications. 202 117239.

[17] Yang J., Li P., Cui Y., Han X., and Zhou M. (2025) Multi-Sensor Temporal Fusion Transformer for Stock Performance Prediction: An Adaptive Sharpe Ratio Approach. Sensors. 25 (3), 976.

[18] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021, May). Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 12, pp. 11106-11115).

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

[20] Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in neural information processing systems, 34, 22419-22430.

[21] Lea, C., Vidal, R., Reiter, A., and Hager, G.D. (2016) Temporal Convolutional Networks: A Unified Approach to Action Segmentation. in: G. Hua, H. Jégou (Eds.), Comput. Vis. – ECCV 2016 Workshop, Springer International Publishing, Champp. 47–54.