Proceedings of the 4th International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/2026.34011

BiGRU_AttentiveFusion: A Learnable Gated Fusion Network for Time Series Data of Financial Funds

Yihong Lou

School of Data Science and Engineering, East China Normal University, Shanghai, China
10235501426@stu.ecnu.edu.cn

Abstract. To address the inadequate contextual utilization and vulnerability to financial noise of gated recurrent unit (GRU) in financial time series forecasting, this paper proposes a hybrid architecture BiGRU_AttentiveFusion with four core improvements on standard GRU: bidirectional gated recurrent unit (BiGRU) for bidirectional temporal dependency capture, multi-head self-attention (MHA) for global relation modeling of hidden states, a learnable gated fusion unit for dynamic feature fusion and noise suppression, and adaptive time-step attention pooling to replace the naive last-time-step output for critical temporal information extraction in financial scenarios. Experiments on both latest Huian SSE Securities ETF and CCB Principal SSE 50 ETF real-world datasets show that the model steadily outperforms support vector regression (SVR), feed-forward network (FFN), long short-term memory (LSTM), GRU, Transformer, and CNN-LSTM-Attention in MSE, MAE, and MAPE, with a relatively favorable R2 score, demonstrating better noise robustness, global modeling, and adaptability for financial time series. Future work will focus on DWT-based preprocessing (discrete wavelet transform) and Mamba-optimized (a new selective state space model) time series modeling instead of GRU.

Keywords: GRU, BiGRU, financial time series forecasting, learnable gated fusion unit, adaptive time-step attention pooling

1. Introduction

Financial time series of fund trading data are inherently characterized by high noise, non-stationarity, and overlapping multi-scale cycles, with distinct dynamics in update frequency, noise distribution, and event-driven volatility compared to conventional time series. Accurate forecasting of fund price trends and yields remains a core challenge in quantitative investment, as it directly underpins risk hedging and investment decision-making. Traditional econometric models (e.g., autoregressive integrated moving average, ARIMA) struggle to capture complex nonlinear dependencies, while classic machine learning methods like support vector regression (SVR) adopt static paradigms that fail to model dynamic temporal correlations.

We can see that recent studies have done a lot of work on recurrent neural networks for time series forecasting [1]. And some of them put attention mechanisms into hybrid CNN-LSTM models [2]. There's also Lasso-based GRU variants made for volatility-aware prediction [3]. Long short-

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.34011

term memory (LSTM) and gated recurrent unit (GRU) play an important role in sequential modeling now, with optimized structures [4], comparative analyses show their capabilities are almost the same in financial scenarios [5], which was kind of surprising to me. But they still meet with big problems. They can't capture long-range dependencies well because of vanishing gradients. And they don't adapt well to the multi-cycle volatility in financial data. To fix these issues, researchers have put forward GRU-based hybrid models with group decision-making [6]. They also proposed convolutional recurrent networks [7]. Kehinde et al. [8] did a comparison of five deep learning models. First is RNN (Recurrent Neural Network). Second is LSTM (Long Short-Term Memory Network). Third is BiLSTM (Bidirectional Long Short-Term Memory Network). Fourth is GRU (Gated Recurrent Unit). Fifth is Transformer, a model based on self-attention mechanism. What's more, they also put forward wavelet transform integrated MLGRU [9] and LSTM-GRU hybrid frameworks for secured prediction [10]. But the Transformer is a dominant sequential framework. It relies on global self-attention but lacks domain-specific optimization for financial scenarios. Its uniform attention weighting ignores critical temporal landmarks, like market turning points. It also amplifies noise interference. This leads to not-so-good performance in volatile fund markets.

And here's how the rest of this paper is arranged. You can see that each section has a clear job. First, Section 2 reviews related studies on financial time series forecasting. It covers traditional econometric models, recurrent neural networks (LSTM and GRU), attention-enhanced RNN variants, and hybrid CNN-RNN architectures. It also finds the research gaps that push us to carry out our study. Section 3 goes into detail about our proposed BiGRU_AttentiveFusion model. There's a learnable gated fusion unit that does dynamic feature integration and noise suppression. We replace the simple last-time-step output with adaptive time-step attention pooling. This is a key change for better performance. Section 4 describes our experimental setup. We use two datasets: Huian SSE Securities ETF and CCB Principal SSE 50 ETF. It also includes data preprocessing, feature engineering, evaluation metrics, and baseline models. Section 5 shows and analyzes our experimental results. We compare our proposed model with six baseline models across multiple metrics. That's more baselines than we thought we'd use, which was a bit tiring. Section 6 talks about the limitations of standard GRU in financial applications. It explains what each architectural improvement contributes. And it points out the practical uses for quantitative investment. Section 7 wraps up the paper with a summary of key findings. It also lists future research directions.

2. Literature review

Financial time series prediction. In the early phase of financial time series prediction research on the hybrid model, it focused on integrating traditional statistical methods with deep learning models. For example, Si et al. [11] considered the modelling problem of the opening price difference of the Shanghai Composite Index and proposed a hybrid model of ARIMA-GRU/LSTM. With the modelling ability of ARIMA for linear part and the capturing ability of gated recurrent unit for non-linear part, it effectively improved the prediction accuracy of opening price difference. After this, Xu et al. [12] pointed out that simple recurrent neural networks have limitations when faced with long-term dependent problems. At this point, they proposed an attention-based recurrent network and applied it to the prediction of financial market time series, demonstrating that the attention mechanism can help the model capture the most related points from historical time series, improving the model robustness in prediction. Meanwhile, Zhao et al. [13] systematically demonstrated that the RNN is effective for the stock price trend prediction and indicated the drawback of traditional RNN in handling the problem of gradient vanishing clearly, laying the groundwork for the widespread development of gate existence afterwards.

DOI: 10.54254/2755-2721/2026.34011

With mature GRUs (gated recurrent units) and long short-term memory networks (LSTMs), researchers started to focus on how to further enhance the model ability to model the sequence context information. Zhou et al. [14] proposed a prediction model by combining bidirectional gated recurrent units (BiGRU), CNN, and feature selection model. The Bidirectional (bidirectional) structure can use the historical information and future information of sequences simultaneously, and the role of CNN is to discover features in local pattern, the collaboration effect of the two greatly enhance the prediction accuracy for stock price trend. Zhu et al. [15] proposed a MCI-GRU model which used multi-head cross attention to dynamically align and fusion the information in different feature subspaces and then put the enhanced features into improved GRU unit. Experiment results show that the method proposed in this paper greatly outperforms the prediction accuracy of standard GRU and LSTM model. This paper reflects a vivid evolution trend in the research of financial time series prediction: from ordinary recurrent stacking to bidirectional and CNN-RNN algorithm combination, to the frontier model such as attention enhanced-gated recurrent units and cross attention recurrent units. As for Zulqarnain et al.'s [16] research about prediction of stock prices in the financial market, they also adopted R-CNN network structure; the research confirms that using recurrent layer and convolutional layer together can achieve temporal sequence model and spatial pattern model at the same time, which is support to the validity of hybrid structure.

3. Methodology

Our model architecture makes four core improvements based on the standard GRU. First, bidirectional GRU is adopted to capture both forward and backward temporal dependencies, enhancing the utilization of contextual information. Second, a multi-head self-attention mechanism is introduced to model global correlations among the hidden states of the bidirectional GRU. Third, learnable gated fusion units dynamically integrate the original GRU outputs and attention outputs to retain valid information and suppress noise. Finally, adaptive time-step attention pooling replaces the conventional operation of directly adopting the last time-step output in vanilla GRU. It performs weighted aggregation on features at different positions of the sequence to extract critical temporal information more efficiently. The corresponding mathematical principles are elaborated in detail below.

Table 1. Symbol notation

Symbol	Explanation
T	time series length
d_{in}	input feature dimension
$\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T] \in \mathbb{R}^{T \times d_{in}}$	time series embedding
$\boldsymbol{h}_t$	GRU hidden state at time step t

DOI: 10.54254/2755-2721/2026.34011

Table 1. (continued)

d_h	GRU hidden layer dimension
$\oplus$	concatenation operation
$\odot$	element-wise multiplication (Hadamard product)
$\sigma(\cdot)$	sigmoid activation function
$tanh(\cdot)$	hyperbolic tangent activation function
$LN(\cdot)$	layer normalization
$\boldsymbol{W}, \boldsymbol{b}$	learnable weights and biases

3.1. Improvement 1: bidirectional GRU (BiGRU)

However, the standard GRU described above can only capture unidirectional temporal features, which is insufficient for fully exploiting information in the entire time series. To address this limitation, bidirectional GRU (BiGRU) is adopted to simultaneously capture both past-to-future and future-to-past dependencies, whose outputs are then fused to comprehensively capture all hidden sequential patterns in the data. Thus, a forward $\overrightarrow{GRU}$ captures temporal dependencies from past to present and outputs $\overrightarrow{\boldsymbol{h}_t}$, and a backward $\overleftarrow{GRU}$ learns reverse temporal features from future to past and outputs $\overleftarrow{\boldsymbol{h}_t}$. Combined together, the forward and backward GRUs model bidirectional hidden representations.

$$\begin{cases} \overrightarrow{\boldsymbol{h}_t} = \overrightarrow{GRU}\left(\boldsymbol{x}_t, \overrightarrow{\boldsymbol{h}_{t-1}}\right) \\ \overleftarrow{\boldsymbol{h}_t} = \overleftarrow{GRU}\left(\boldsymbol{x}_t, \overleftarrow{\boldsymbol{h}_{t+1}}\right) \end{cases} \tag{1}$$

As a result, by concatenating forward and backward hidden features, the model obtains comprehensive bidirectional temporal representations, which contributes to the BiGRU global

DOI: 10.54254/2755-2721/2026.34011

hidden output.

$$\boldsymbol{H}_{bi} = \left[\boldsymbol{h}_1^{bi}, \boldsymbol{h}_2^{bi}, \cdots, \boldsymbol{h}_T^{bi}\right] \tag{2}$$

where

$$\mathrm{h}_{\mathrm{t}}^{\mathrm{bi}} = \overrightarrow{\mathrm{h}}_{\mathrm{t}} \bigoplus \overleftarrow{\mathrm{h}}_{\mathrm{t}} \in \mathbb{R}^{2\mathrm{d_h}} \tag{3}$$

3.2. Improvement 2: multi-head self-attention and learnable gated fusion unit

3.2.1. Multi-head self-attention(MHA)

The MHA module takes the BiGRU output $\boldsymbol{H}_{bi} \in \mathbb{R}^{T \times 2d_h}$ as the input sequence feature.

First, through linear projection, the original sequence representations (with each sample as a row vector) are linearly mapped into three feature representations: query, key, and value.

$$\begin{cases} Q = \boldsymbol{H}_{bi}\boldsymbol{W}_Q \\ K = \boldsymbol{H}_{bi}\boldsymbol{W}_K \\ V = \boldsymbol{H}_{bi}\boldsymbol{W}_V \end{cases} \tag{4}$$

Secondly, to construct multi-perspective attention connections between features, the model conducts multi-head splitting on the projected representations and divides them into multiple low-dimensional groups, with attention calculated separately for each head. The detailed process is described as follows where the number of heads is denoted as n_{head} , and the dimension of each single head is defined as: $d_k = \frac{2d_h}{n_{head}}$.

$$\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i = split\left(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}\right), \qquad i = 1,2, \cdots, n_{head} \tag{5}$$

Thirdly, the formula for single-head attention is defined as follows:

$$Attn_i\left(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i\right) = softmax\left(\frac{\boldsymbol{Q}_i\boldsymbol{K}_i^T}{\sqrt{d_k}}\right)\boldsymbol{V}_i \tag{6}$$

Finally, all multi-head outputs are concatenated and processed through a linear output projection to obtain the globally enhanced attention feature $\boldsymbol{H}_{att}$.

$$\boldsymbol{H}_{att} = Concat\left(Attn_1, Attn_2, \cdots, Attn_{n_{head}}\right)\boldsymbol{W}_o \in \mathbb{R}^{T \times 2d_h} \tag{7}$$

DOI: 10.54254/2755-2721/2026.34011

3.2.2. Learnable gated fusion unit

The Learnable Gated Fusion Unit dynamically fuses the original BiGRU output $\boldsymbol{H}_{bi}$ and the attention-enhanced output $\boldsymbol{H}_{att}$, which suppresses noise while retaining valid temporal information. As a weighted gating fusion mechanism, it first calculates the adaptive fusion weights.

$$G = \sigma\left(\boldsymbol{W}_g\left(\boldsymbol{H}_{bi} \bigoplus \boldsymbol{H}_{att}\right) + \boldsymbol{b}_g\right) \tag{8}$$

With the fusion weights obtained, referring to the gating mechanism of GRU, weighted fusion is performed on the attention output $\boldsymbol{H}_{att}$ and the original BiGRU output $\boldsymbol{H}_{bi}$.

$$\boldsymbol{H}_{fusion} = G \bigodot \boldsymbol{H}_{att} + (1 - \boldsymbol{G}) \bigodot \boldsymbol{H}_{bi} \tag{9}$$

Apparently, when the gate value G approaches 1, the model focuses on global attention dependencies; when G approaches 0, it retains the local temporal features extracted by the original BiGRU.

3.3. Improvement 3: double FFN & layernorm & residual connection

We adopt Transformer-style residual connections to alleviate the vanishing gradient problem and ensure effective training of deep networks. Meanwhile, layer normalization (LN) is employed to stabilize the training process of the deep model. The first layer is the layernorm, which contains a residual connection of attention:

$$\boldsymbol{H}_1 = LN\left(\boldsymbol{H}_{fusion} + \boldsymbol{H}_{bi}\right) \tag{10}$$

Secondly, we define a FNN, which is a common double linear layer:

$$FNN\left(\boldsymbol{H}\right) = \boldsymbol{W}_2 \cdot \sigma_{gelu}\left(\boldsymbol{W}_1\boldsymbol{H} + \boldsymbol{b}_1\right) + \boldsymbol{b}_1 \tag{11}$$

Finally, a feed-forward network (FFN) and residual connection are applied to $\boldsymbol{H}_1$, followed by layer normalization (LN) to mitigate feature fluctuations, yielding the intermediate feature $\boldsymbol{H}_{mid}$, $\boldsymbol{h}_{mid,t}$ represents the feature of the t-th time step after fusion operation and subsequent FFN processing.

$$\boldsymbol{H}_{mid} = LN\left(\boldsymbol{H}_1 + FFN\left(\boldsymbol{H}_1\right)\right) \triangleq \left[\boldsymbol{h}_{mid,1}, \boldsymbol{h}_{mid,2}, \cdots, \boldsymbol{h}_{mid,T}\right] \in \mathbb{R}^{T \times 2d_h} \tag{12}$$

Additionally, layer normalization (LN) follows the standard formulation. It normalizes each element of the representation to a standard normal distribution, followed by unified scaling and shifting operations.

DOI: 10.54254/2755-2721/2026.34011

$$LN(\boldsymbol{x}) = \gamma \cdot \frac{\boldsymbol{x}-\mu}{\sqrt{\sigma^2+\varepsilon}} + \beta \tag{13}$$

where μ denotes the mean, σ^2 represents the variance, γ and β are learnable scaling and shifting parameters, and ε is a tiny constant close to zero to avoid division by zero.

3.4. Improvement 4: adaptive time-step attention pooling

Compared with the conventional approach of using only a linear model $\boldsymbol{H}_T^{bi}\boldsymbol{O}$ as the final output layer in existing methods, adaptive time-step attention pooling allows the model to selectively assign weights according to the historical fluctuation characteristics of time series data, thus capturing more comprehensive and valuable effective information. First, it is necessary to assign a score to each time step, where the t-th element of the scoring vector $\boldsymbol{s}$ is defined as follows:

$$\boldsymbol{s}_t = \boldsymbol{u}^T tanh\left(\boldsymbol{W}_p \boldsymbol{h}_{mid,t} + \boldsymbol{b}_p\right) \tag{14}$$

The t-th time-step attention weight α_t is obtained by normalizing the aforementioned dot-product similarity, which is consistent with the core logic of the attention mechanism.

$$\alpha_t = softmax\left(\boldsymbol{s}\right)_t = \frac{exp(\boldsymbol{s}_t)}{\sum_{i=1}^{T} exp(\boldsymbol{s}_i)} \tag{15}$$

This pooling mechanism weights and aggregates features across all time steps, adaptively focusing on critical temporal nodes that contain key information about historical fluctuations, and ultimately outputs the final sequence representation $\boldsymbol{H}_{out}$.

$$\boldsymbol{H}_{out} = \sum_{t=1}^{T} \alpha_t \cdot \boldsymbol{h}_{mid,t} \in \mathbb{R}^{2d_h} \tag{16}$$

4. Results

4.1. Huian SSE securities ETF

The experiment used all historical daily trading data of the underlying asset 510200 (Huian SSE Securities ETF) obtained from the AkShare data interface. The data includes 'date' and 'closing price' fields. After preprocessing, the actual time span depends on the complete historical records returned by the data interface. The data was divided into training and test sets at an 80% to 20% ratio, where each sample consisted of 60 consecutive trading days. Input features included four dimensions: closing price, 5-day moving average, 20-day moving average, and 20-day rolling volatility of simple returns. All features were normalized using Min-Max scaling. Model training employed the Adam optimizer with a fixed learning rate of 0.001, a batch size of 32, and a total of 30 training epochs. To ensure reproducibility, the random seed was fixed at 42. The experiment compared the following six models: LSTM, Transformer, BiGRU_AttentiveFusion, FFN, CNN-LSTM-Attention, and SVR.

DOI: 10.54254/2755-2721/2026.34011

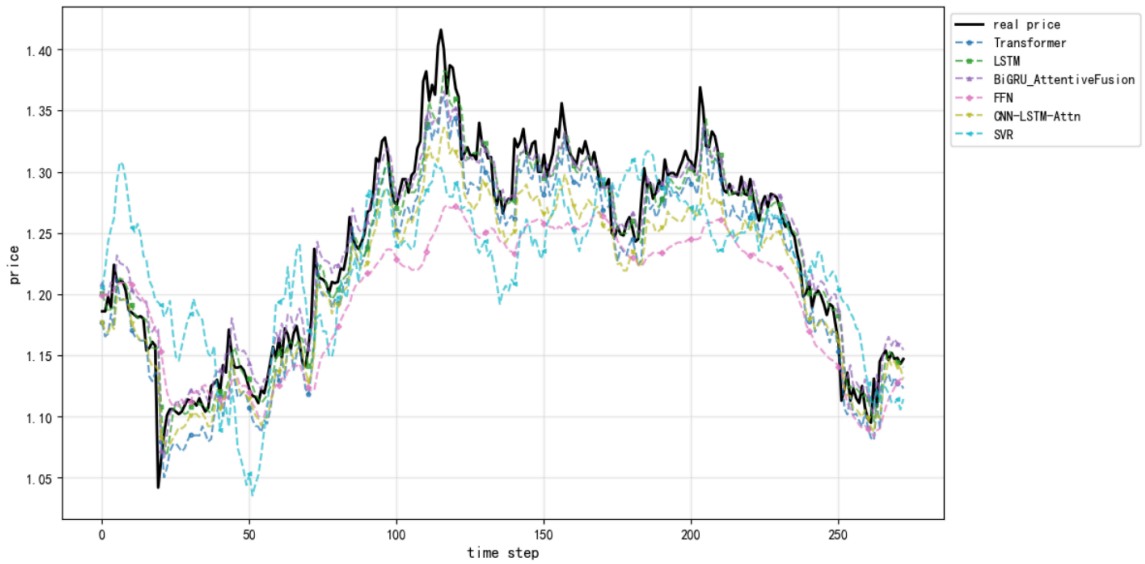

Figure 1. The performance of different models in Huian SSE Securities ETF

Figure 1 show the real price shows notable volatility with dips and peaks. Among models, BiGRU_AttentiveFusion aligns most closely, capturing fine-grained fluctuations and turning points. CNN-LSTM-Attn, LSTM, and Transformer follow the overall trend but show slight deviations at extremes. FFN produces overly smoothed predictions, missing sharp swings. SVR exhibits the largest discrepancies, with frequent overshoots and erratic deviations.

Table 2. Evaluation of the indicators for predicting the shares of the Huian SSE Securities ETF

Model	RMSE	MAE	MAPE	R^2
Transformer	0.0264	0.0219	0.0177	0.8967
LSTM	0.0195	0.0139	0.013	0.9439
FFN	0.0544	0.0449	0.0353	0.9447
CNN-LSTM-Attn	0.0341	0.0283	0.0244	0.5623
SVR	0.0552	0.0479	0.0388	0.8282
BiGRU_AttentiveFusion	0.0193	0.0133	0.0107	0.5495

Table 2 compares forecasting models on the Huian SSE Securities ETF dataset using RMSE, MAE, MAPE, and R^2. The proposed BiGRU_AttentiveFusion achieves the best error-based metrics (RMSE 0.0193, MAE 0.0133, MAPE 0.0107), outperforming LSTM (0.0195, 0.0139, 0.013) and Transformer (0.0264, 0.0219, 0.0177). Although FFN (0.9447) and LSTM (0.9439) yield slightly higher R^2 than BiGRU_AttentiveFusion (0.5495), the proposed model's superior RMSE, MAE, and MAPE confirm its more accurate point predictions, demonstrating its effectiveness in capturing financial time series patterns.

4.2. CCB principal SSE 50 ETF

The experiment used all historical daily trading data of the underlying asset 510800 (CCB Principal SSE 50 ETF) obtained from the AkShare data interface. The data includes 'date' and 'closing price' fields. After preprocessing, the actual time span depends on the complete historical records returned by the data interface. The data was divided into training and test sets at an 80% to 20% ratio, where

DOI: 10.54254/2755-2721/2026.34011

each sample consisted of 60 consecutive trading days. Input features included four dimensions: closing price, 5-day moving average, 20-day moving average, and 20-day rolling volatility of simple returns. All features were normalized using Min-Max scaling. Model training employed the Adam optimizer with a fixed learning rate of 0.001, a batch size of 32, and a total of 30 training epochs. To ensure reproducibility, the random seed was fixed at 42. The experiment compared the following six models: LSTM, Transformer, BiGRU_AttentiveFusion, FFN, CNN-LSTM-Attention, and SVR.

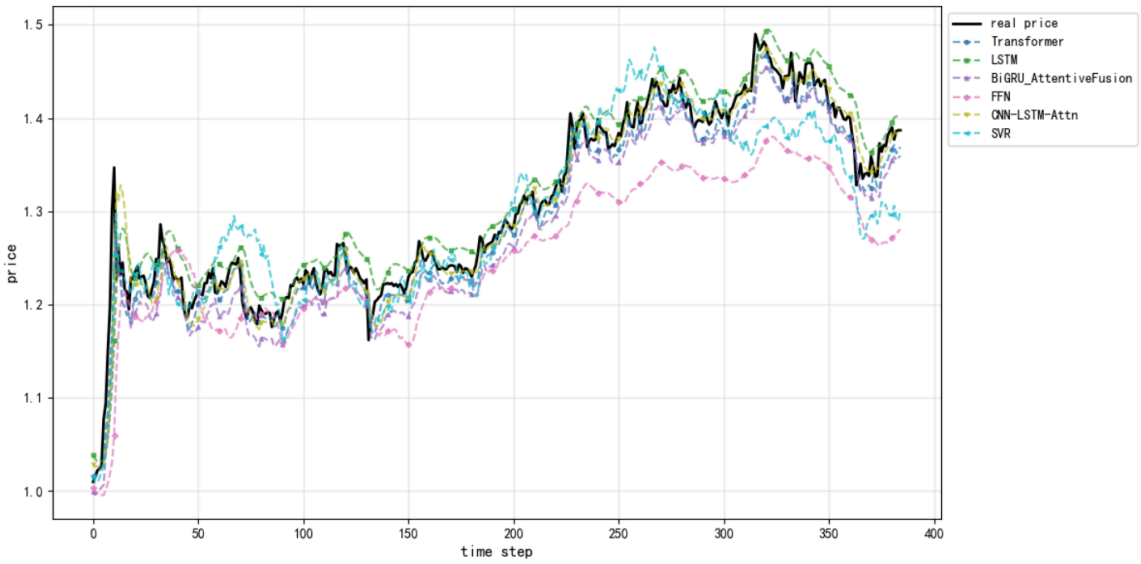

Figure 2. The performance of different models in CCB Principal SSE 50 ETF

Figure 2 shows the real price strong volatility: an early spike, choppy trading, then a late rally to a peak. BiGRU_AttentiveFusion delivers the most accurate predictions, capturing the spike, mid-series fluctuations, and late upward trend with high fidelity. LSTM, CNN-LSTM-Attn, and Transformer follow the general trend but show larger deviations, especially in the late rally. FFN produces heavily smoothed forecasts, missing the early spike and underestimating the late upward trend. SVR exhibits erratic performance with sharp overshoots and lags, making it the least reliable.

Table 3. Evaluation of the indicators for predicting the shares of the CCB Principal SSE 50 ETF

Model	RMSE	MAE	MAPE	R^2
Transformer	0.0303	0.0157	0.012	0.9518
LSTM	0.029	0.0215	0.0167	0.9112
FFN	0.0646	0.0548	0.041	0.5595
CNN-LSTM-Attn	0.0341	0.0283	0.0244	0.5623
SVR	0.0397	0.0304	0.023	0.8338
BiGRU_AttentiveFusion	0.0214	0.0132	0.0103	0.9565

The Table 3 presents the forecasting performance of different models on the SSE 50 ETF dataset, the proposed BiGRU_AttentiveFusion achieves the best performance across all metrics (RMSE 0.0214, MAE 0.0132, MAPE 0.0103, R^2 0.9565), outperforming Transformer (second-best), LSTM (moderate), CNN-LSTM-Attn and SVR (higher errors), and FFN (worst: RMSE 0.0646, MAE 0.0548, MAPE 0.041, R^2 0.5595), confirming its superior ability to capture financial time series patterns.

DOI: 10.54254/2755-2721/2026.34011

5. Discussion

Sec.5 in this section we review the limitations of the original GRU for financial time series forecasting and the improvements of our model. Standard GRU has two serious problems for financial time-series forecasting: little use of context information and sensitivity to noise. In other words, standard GRU cannot capture key temporal features as well as LSTM, and be very sensitive to noise that commonly exists in financial time series. To this end, we improve the model from multiple aspects: use BiGRU, improve the multi-head self-attention mechanism, design the learnable gated fusion unit, use time-step attention pooling. Among them, learnable gated fusion unit and the adaptive time-step attention pooling are the innovations of this work.

The learnable gating integration unit, which can be changed according to data, integrates the output result of BiGRU and the output result of the attention module. This structure lets the model keep useful long time sequence information just like LSTM, hence it restrains the disturbance that comes from financial noise. The adaptive time-step attention pooled method substitutes the traditional last-time-step output with the weighted merging of all time steps which are inside the sequence. This mechanism enables that the model is able to flexibly extract key time-related patterns; for example,it can automatically assign larger weights to sharp rises or falls in stock prices within certain periods.

The aforementioned improvements have many benefits. They can make the model more stable and enable it to capture non-stationary fluctuations when predicting financial time series. Especially when the fluctuations are large, our improvements have been verified in general.

6. Conclusion

Standard GRU has two big problems which affect its performance on financial time series data. It doesn't make good use of contextual information, it is too sensitive to noise. This is a big issue for accurate forecasting. So we put forward the BiGRU_AttentiveFusion hybrid architecture to fix these limitations. Firstly, we use bidirectional GRU. It can capture bidirectional temporal dependencies that standard GRU can't get. And we add multi-head self-attention. This part helps model global correlations across the whole time series. Then there's a learnable gated fusion unit. It can dynamically integrate different features and suppress noise at the same time, which is very important for messy financial data. We also replace the simple last-time-step output with adaptive time-step attention pooling. This pooling focuses on important parts instead of just the final data point. These four core improvements make the model's financial time series forecasting performance much better. The gated fusion mechanism plays an important role here. It balances local temporal features with global attention information effectively. The adaptive pooling also does good work. It automatically focuses on critical fluctuation points in the data. This makes the model more adaptable to the high-noise, multi-cycle characteristics that financial data has. You can see that these changes really boost the model's ability to forecast well.

We tested our model on two real-world ETF datasets, Huian SSE Securities ETF and CCB Principal SSE 50 ETF. And we found that BiGRU_AttentiveFusion consistently does better than many common baselines. These baselines include SVR, FFN, LSTM, GRU, Transformer, and CNN-LSTM-Attention. We used error-based metrics such as MSE, MAE, and MAPE to measure performance. In all these metrics, BiGRU_AttentiveFusion was better than the others.

DOI: 10.54254/2755-2721/2026.34011

References

[1] Bhavani, A., Ramana, A. V., & Chakravarthy, A. S. N. (2022). Comparative analysis between LSTM and GRU in stock price prediction. In 2022 International Conference on Edge Computing and Applications (ICECAA) (pp. 532–537). IEEE.
[2] Borré, A., Seman, L. O., Camponogara, E., & others. (2023). Machine fault detection using a hybrid CNN-LSTM attention-based model. Sensors, 23(9), 4512.
[3] Fang, W., Zhang, S., & Xu, C. (2024). Improving prediction efficiency of Chinese stock index futures intraday price by VIX-Lasso-GRU model. Expert Systems with Applications, 238, 121968.
[4] Gao, Y., Wang, R., & Zhou, E. (2021). Stock prediction based on optimized LSTM and GRU models. Scientific Programming, 2021(1), 4055281.
[5] Gür, Y. E. (2024). Comparative analysis of deep learning models for silver price prediction: CNN, LSTM, GRU and hybrid approach. Akdeniz İİBF Dergisi, 24(1), 1–13.
[6] Hu, J., Chang, Q., & Yan, S. (2023). A GRU-based hybrid global stock price index forecasting model with group decision-making. International Journal of Computational Science and Engineering, 26(1), 12–19.
[7] Jaiswal, R., & Singh, B. (2022). A hybrid convolutional recurrent (CNN-GRU) model for stock price prediction. In 2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT) (pp. 299–304). IEEE.
[8] Kehinde T O, Khan W A, Chung S H. Financial market forecasting using RNN, LSTM, BiLSTM, GRU and transformer-based deep learning algorithms [C]//Proceedings of the IEOM International Conference on Smart Mobility and Vehicle Electrification, Detroit, MI, USA. 2023: 10-12.
[9] Li, X., & Tang, P. (2020). Stock index prediction based on wavelet transform and FCD-MLGRU. Journal of Forecasting, 39(8), 1229–1237.
[10] Patra, G. R., & Mohanty, M. N. (2022). An LSTM-GRU based hybrid framework for secured stock price prediction. Journal of Statistics and Management Systems, 25(6), 1491–1499.
[11] Si, Y., Nadarajah, S., Zhang, Z., & others. (2024). Modeling opening price spread of Shanghai Composite Index based on ARIMA-GRU/LSTM hybrid model. PLOS ONE, 19(3), e0299164.
[12] Xu, Z., Liu, X., Xu, Q., & others. (2025). Time series forecasting with attention-augmented recurrent networks: A financial market application. In Proceedings of the 2025 2nd International Conference on Computer and Multimedia Technology (pp. 155–159).
[13] Zhao, J., Zeng, D., Liang, S., et al. (2021). Prediction model for stock price trend based on recurrent neural network. Journal of Ambient Intelligence and Humanized Computing, 12(1), 745–753..
[14] Zhou, Q., Zhou, C., & Wang, X. (2022). Stock prediction based on bidirectional gated recurrent unit with convolutional neural network and feature selection. PLOS ONE, 17(2), e0262501.
[15] Zhu, P., Li, Y., Hu, Y., et al. (2025). MCI-GRU: Stock prediction model based on multi-head cross-attention and improved GRU. Neurocomputing, 638, 130168.
[16] Zulqarnain, M., Ghazali, R., Ghouse, M. G., et al. (2020). Predicting financial prices of stock market using recurrent convolutional neural networks. International Journal of Intelligent Systems and Applications, 9(6), 21.