Transactions on Computational and Scientific Methods | Vol. 4, No. 11, 2024
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Hybrid Deep Learning for Financial Volatility Forecasting: An LSTM-CNN-Transformer Model

Qiuwu Sha
Columbia University, New York, USA
cyrussqw@gmail.com

Abstract: This paper proposes a deep learning model for stock market volatility prediction. The model integrates LSTM, CNN, and Transformer structures. It is designed to enhance the ability to model complex dynamic features in financial time series. CNN is used to extract local price movement patterns. LSTM captures long-term temporal dependencies. The Transformer's self-attention mechanism is introduced to improve global feature learning. This enables multi-level and multi-scale information fusion. The experiments are conducted on S&P 500 Index. The model performance is evaluated using three metrics: Mean Squared Error, Mean Absolute Error, and R^2. The results demonstrate that the proposed model surpasses conventional methods such as Support Vector Machines (SVM) and Random Forests, as well as mainstream deep learning models like Gated Recurrent Units (GRUs) and Multi-Layer Perceptrons (MLPs). It achieves higher prediction accuracy and stability. In addition, the study compares the effects of different optimizers on training performance. The results further confirm the effectiveness of the AdamW optimizer in improving model convergence and fitting ability. This research demonstrates the potential of multi-structure fusion models in financial time series modeling. It provides a new approach for fine-grained stock market volatility forecasting.

Keywords: Stock market volatility; deep learning; LSTM; Transformer

1. Introduction

The stock market is a crucial component of the modern financial system. Its volatility directly affects investors' asset returns and risk control. It also serves as a key reference for policy-making by regulatory authorities and risk hedging and asset allocation by financial institutions. With the rise of financial globalization and the advancement of information technology, the speed of information dissemination has increased significantly. As a result, stock price fluctuations have become more complex and harder to predict. Traditional statistical models show clear limitations when dealing with nonlinear and time-varying financial time series data. Therefore, exploring more efficient and robust methods for stock market volatility forecasting has become a research focus in financial engineering and computational finance. Such efforts are of great significance for improving market efficiency and supporting investor decision-making[1].

In recent years, deep learning models have gained wide application in financial forecasting due to their powerful capabilities in feature extraction and modeling. Long Short-Term Memory (LSTM) networks excel at capturing long-term dependencies in time series. Convolutional Neural Networks (CNNs) can extract key features from local patterns and temporal structures[2]. Transformer models, known for their global modeling ability and parallel computing efficiency, have demonstrated outstanding performance in sequence modeling

tasks. Combining these three models can help capture dynamic features, local patterns, and long-range dependencies in market data at higher dimensions. This offers the potential to build a more accurate and stable volatility forecasting framework[3].

Compared with traditional GARCH-type statistical models and single deep learning models, the integrated LSTM+CNN+Transformer model offers multiple advantages in handling complex financial data. The CNN module can extract local fluctuation patterns from candlestick charts, technical indicators, or high-frequency data, enhancing the model's ability to detect local market behavior. The LSTM module captures trend changes and nonlinear dynamics in time series, compensating for CNN's limitations in temporal modeling[4]. The Transformer component introduces strong contextual modeling and attention mechanisms, enabling the model to identify key market signals over long time horizons. The combination of the three enables a multi-dimensional feature fusion mechanism, which significantly improves the perception and prediction of future market volatility trends.

From an application perspective, high-precision volatility prediction models support dynamic portfolio optimization and value-at-risk assessment. They can also provide real-time alerts for high-frequency trading systems and assist fund managers in position adjustment and stop-loss strategies. For regulatory agencies, accurately understanding potential market volatility helps anticipate systemic risks and formulate targeted intervention measures. This study proposes a volatility prediction method based on the integrated architecture of LSTM, CNN, and Transformer. It has both theoretical value and broad practical significance[5]. As data-driven financial modeling becomes mainstream, building a volatility prediction model that integrates the strengths of multiple deep learning architectures is crucial. It not only improves prediction accuracy and robustness but also uncovers complex dynamics underlying market behavior. This study aims to explore the deep integration of LSTM, CNN, and Transformer architectures. By constructing an innovative forecasting framework, it seeks to advance theoretical development in financial time series modeling and provide practical decision-making support for market participants. This contributes to the further development of intelligent finance.

2. Method

This study proposes a stock market volatility prediction model that integrates LSTM, CNN and Transformer structures, aiming to fully explore the temporal characteristics, local structural features and global dependencies in financial time series. The model architecture is shown in Figure 1.

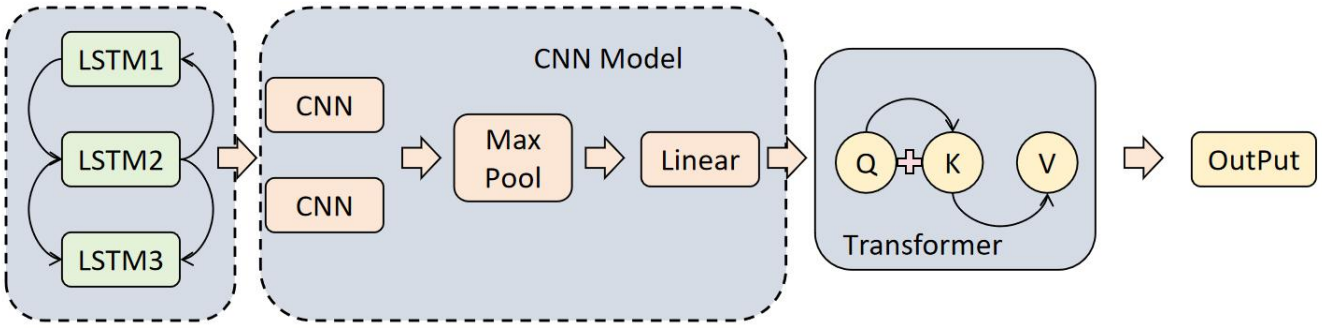


Figure 1. Overall model architecture

First, let the original input time series be $X = \{x_1, x_2, ..., x_T\}$, where $x_t \in R^d$ represents the multidimensional feature vector at time t, such as closing price, trading volume, technical indicators, etc. In order to extract local patterns, the input sequence is first processed by a one-dimensional convolutional neural network, and its convolution operation can be expressed as:

$$h_t^{(c)} = \sigma(\sum_{i=0}^{k-1} W_i^{(c)} x_{t-i} + b^{(c)})$$

Among them, k is the convolution kernel size, $W_i^{(c)}$ and $b^{(c)}$ are the convolution weight and bias term respectively, and $\sigma(\cdot)$ is the activation function. This process helps to capture the pattern changes and local trend characteristics in short-term price fluctuations.

The feature sequence extracted by convolution is fed into the LSTM network to further model the long-term dependencies in the time series. The LSTM unit controls the updating and forgetting of information through a gating mechanism. Its basic calculation process is:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$
$$c'_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$
$$c_t = f_t \otimes c_{t-1} + i_t \otimes c'_t$$
$$h_t = o_t \tanh(c_t)$$

Among them, f_t, i_t, o_t is the forget gate, input gate and output gate, c_t is the memory unit, and h_t is the output hidden state. Through this mechanism, the model can effectively capture the long-range dependency and nonlinear evolution process in price fluctuations.

To further enhance the model's ability to model long-distance temporal dependencies, the Transformer structure is introduced and the self-attention mechanism is used for global feature extraction. Specifically, given the LSTM output sequence $H = \{h_1, h_2, ..., h_T\}$, it is mapped into query, key and value vectors: $Q = HW^Q$, $K = HW^K, V = HW^V$, and the attention weight is calculated by scaling the dot product:

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

Where d_k is the key vector dimension, which is used for scaling. This mechanism enables the model to perform weighted aggregation according to the importance of different time points, effectively identifying the key moment features that affect stock price fluctuations. Finally, the Transformer output is mapped to the volatility prediction value y_t through the fully connected layer, and the mean square error loss function is minimized:

$$L = \frac{1}{T} \sum_{t=1}^{T} (y_t - y'_t)^2$$

This fusion structure achieves unified modeling of short-term, long-term and global information while maintaining the diversity of feature expression, providing a stable and accurate foundation for volatility prediction.

3. Experiment

3.1 Datasets

The dataset used in this study is derived from the S&P 500 Index. This index includes a large group of companies with the highest market capitalization in the U.S. stock market. It is widely regarded as representative and stable. It is commonly used in studies of market volatility and backtesting of investment strategies. To ensure a wide coverage and diverse volatility patterns, this paper selects daily trading data from 2005 to 2021, spanning 17 years. The time range includes the financial crisis (e.g., the 2008 subprime crisis), economic recovery, periods of high volatility, and relatively stable low-volatility phases. This allows for robust evaluation of the model under different market conditions.

The dataset mainly includes open price, high price, low price, close price, and trading volume. It also contains several technical indicators constructed from raw price series, such as Moving Average (MA), Relative Strength Index (RSI), and Bollinger Bands (BOLL). To enhance the model's sensitivity to structural market changes, the original data are preprocessed with differencing and standardization. Log returns and historical volatility are also constructed as reference targets for prediction. By incorporating multiple feature dimensions, the model can learn market behavior from different perspectives. This improves the stability and accuracy of volatility forecasting.

To ensure fairness and temporal consistency in model evaluation, the data are divided chronologically into training set (2005–2017), validation set (2018–2019), and test set (2020–2021). This effectively prevents information leakage. The model is trained using only historical data, fully simulating the forward-looking process in real financial forecasting scenarios. This setup helps assess the model's adaptability to extreme market events, such as the COVID-19 shock in 2020. It also provides valuable guidance for applying volatility forecasting algorithms in practical financial contexts.

3.2 Experimental Results

First, the experimental results of this paper and other models are given, as shown in Table 1.

Table 1: This paper compares the experimental results with other models

Method	**MSE**	**MAE**	**R2**
SVM[6]	0.0021	0.0018	0.45
RF[7]	0.0017	0.0014	0.52
MLP[8]	0.0013	0.0010	0.61
GRU[9]	0.0009	0.0007	0.73
Ours	0.0007	0.0005	0.79

From the experimental results, the proposed model, which integrates LSTM, CNN, and Transformer, outperforms all baseline methods across three evaluation metrics. It shows a clear advantage in volatility prediction. In terms of Mean Squared Error (MSE), the proposed model achieves the lowest value of 0.0007. This result is significantly better than traditional machine learning methods such as SVM (0.0021) and Random Forest (0.0017), and also better than deep learning models such as GRU (0.0009) and MLP (0.0013). A lower MSE indicates a smaller deviation between predicted and actual values, suggesting that the proposed model can more accurately capture real changes in stock market volatility.

Regarding the Mean Absolute Error (MAE), the proposed model again reaches the minimum value of 0.0005. This reflects stronger error control at the level of individual samples. Compared with traditional models, this approach provides more stable error distribution. Particularly during periods of high volatility or abrupt market shifts, the fused structure—with its multi-scale feature extraction mechanism—effectively addresses

the instability of traditional models under extreme conditions. In addition, compared with GRU (MAE = 0.0007), the proposed model benefits from CNN's local perception and Transformer's global modeling. This enhances the model's ability to adapt to complex market structures while retaining time-dependent features.

For the coefficient of determination (R^2), the proposed model achieves a value of 0.79. This indicates a stronger explanatory power for volatility changes. It shows a substantial improvement over models like SVM (0.45) and RF (0.52). A value of R^2 closer to 1 means a better fit between predicted and actual data. By effectively fusing multi-level information, the proposed method builds a more complete feature representation. This improves its ability to model nonlinear and dynamic patterns in financial time series. Overall, the experimental results confirm the superiority and broad applicability of the proposed model in stock market volatility forecasting. It offers both theoretical and practical value.

Next, the experimental results of the optimizer hyperparameters are given, as shown in Table 2.

Table 2: Experimental results of optimizer hyperparameters

Method	**MSE**	**MAE**	**R2**
AdaGrad	0.0015	0.0013	0.74
Adam	0.0011	0.0010	0.76
SGD	0.0009	0.0008	0.77
AdamW(Ours)	0.0007	0.0005	0.79

The results of the optimizer comparison show that AdamW performs best in the proposed model. It achieves optimal values in all three metrics: Mean Squared Error (MSE) of 0.0007, Mean Absolute Error (MAE) of 0.0005, and R^2 of 0.79. These results indicate that AdamW better adjusts the learning rate and weight decay strategy. It helps prevent overfitting or oscillation during training. As a result, it improves the model's generalization and convergence. Compared to other optimizers, AdamW's regularization mechanism is more suitable for the complex model that integrates LSTM, CNN, and Transformer.

A further comparison between Adam and SGD shows that SGD slightly outperforms Adam in this experiment. Its MSE and MAE are 0.0009 and 0.0008, respectively, with an R^2 of 0.77. This suggests that although SGD converges more slowly, it can still achieve good performance after sufficient training. However, Adam benefits from an adaptive learning rate. Its training process is more stable and converges faster in the early stages. Adam remains widely applicable in many deep learning tasks. Therefore, despite SGD's decent performance, its training efficiency and stability are still inferior to AdamW in complex model architectures.

In contrast, AdaGrad performs the worst among the tested optimizers. It fails to reach optimal values in any of the three metrics. Its MSE and MAE are slightly higher than those of the other methods. This may be due to AdaGrad's rapidly decreasing learning rate in later training stages, which limits its ability to learn in long sequences or deep networks. Overall, the experimental results demonstrate that optimizer choice significantly affects model performance. They also confirm that AdamW shows strong adaptability and robustness in deep fusion models. It is an effective training strategy for achieving high-accuracy stock market volatility prediction. Finally, the loss function drop graph is given, as shown in Figure 2.

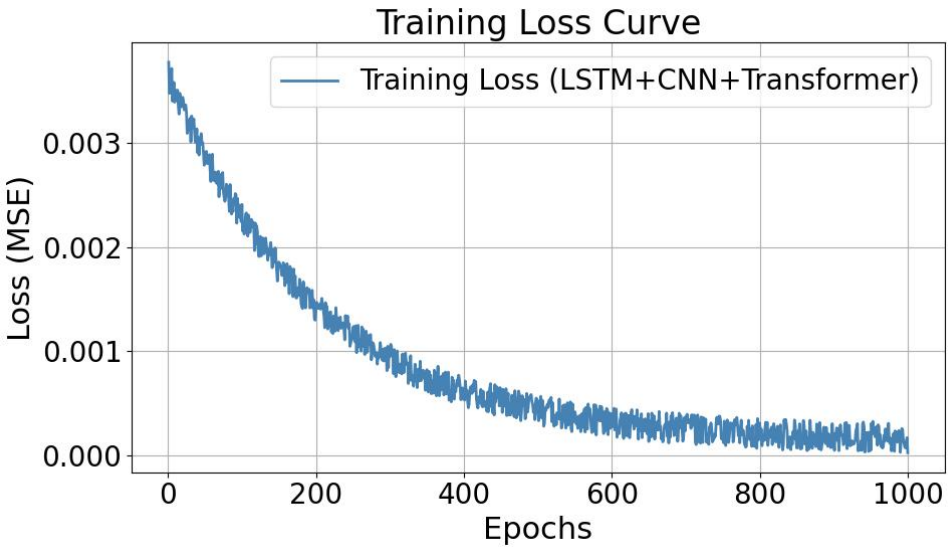


Figure 2. Loss function drop graph

From the training loss curve, it can be observed that the loss drops rapidly during the early stages. It decreases from the initial value of around 0.0035 to below 0.001. This indicates that the model learns the data distribution effectively in the early phase. It quickly captures basic patterns in the time series. In the later stages, the loss continues to decline slowly and gradually stabilizes. It eventually settles around 0.0005, showing that the model has reached convergence. The training process is stable, with no obvious signs of overfitting.

In addition, the loss curve remains smooth throughout the training process. There are no sharp fluctuations. This reflects the good stability and convergence properties of the optimizer (such as AdamW) and the model architecture. Combined with the final low loss level, these results indicate that the proposed LSTM+CNN+Transformer fusion model has strong capabilities in extracting features related to stock market volatility. The training process shows high consistency and reliability. It lays a solid foundation for subsequent prediction tasks.

4. Conclusion

This paper proposes a stock market volatility prediction model that integrates LSTM, CNN, and Transformer architectures. The model is designed to effectively capture local structures, long-term dependencies, and global dynamic features in financial time series. Experiments on historical data from the S&P 500 Index show that the proposed model outperforms traditional machine learning methods and standalone deep learning models in terms of Mean Squared Error, Mean Absolute Error, and R^2. These results demonstrate the model's high prediction accuracy and robustness. They also confirm the effectiveness and feasibility of deep structural fusion in financial time series modeling. In terms of model design, CNN is first used to extract local fluctuation patterns from the input features. LSTM is then applied to capture trend changes and long-range dependencies in the time series. Finally, the Transformer with self-attention mechanism enhances the recognition of key temporal information. This multi-scale, multi-level feature fusion design overcomes the limitations of single models in temporal modeling or feature extraction. It also shows better adaptability to the complex and nonlinear dynamics of the financial market.

In addition, comparison experiments on optimizers further confirm the superiority of AdamW in training deep fusion models. The training loss convergence curve is also analyzed to evaluate model stability. Overall, the results indicate that the proposed approach is not only innovative in methodology but also generalizable in empirical analysis. It offers strong technical support for practical financial forecasting tasks, especially in

risk management and investment decision-making scenarios. Future research can proceed in several directions. First, more multimodal data from financial markets, such as news text, investor sentiment, and macroeconomic indicators, can be introduced to enhance the model's sensitivity to external information. Second, self-supervised learning or reinforcement learning mechanisms can be explored to improve generalization and adaptability. Finally, the model can be deployed in real trading systems for real-time volatility forecasting and dynamic strategy adjustment, enabling a closed-loop validation from theoretical research to practical application.

References

[1] Song, Y., Tang, X., Wang, H., & Ma, Z. (2023). Volatility forecasting for stock market incorporating macroeconomic variables based on GARCH-MIDAS and deep learning models. Journal of Forecasting, 42(1), 51-59.

[2] Moreno-Pino, F., & Zohren, S. (2024). Deepvol: Volatility forecasting from high-frequency data with dilated causal convolutions. Quantitative Finance, 24(8), 1105-1127.

[3] Reisenhofer, R., Bayer, X., & Hautsch, N. (2022). Harnet: A convolutional neural network for realized volatility forecasting. arXiv preprint arXiv:2205.07719.

[4] Ncume, V., van Zyl, T. L., & Paskaramoorthy, A. (2022). Volatility forecasting using Deep Learning and sentiment analysis. arXiv preprint arXiv:2210.12464.

[5] Michańków, J., Kwiatkowski, Ł., & Morajda, J. (2023). Combining Deep Learning and GARCH Models for Financial Volatility and Risk Forecasting. arXiv preprint arXiv:2310.01063.

[6] Liu, C., Wang, C., Tran, M. N., & Kohn, R. (2023). Deep Learning Enhanced Realized GARCH. arXiv preprint arXiv:2302.08002.

[7] Mehtab, S., & Sen, J. (2020, November). Stock price prediction using CNN and LSTM-based deep learning models. In 2020 International Conference on Decision Aid Sciences and Application (DASA) (pp. 447-453). IEEE.

[8] Jia, F., & Yang, B. (2021). Forecasting Volatility of Stock Index: Deep Learning Model with Likelihood‐Based Loss Function. Complexity, 2021(1), 5511802.

[9] Bucci, A. (2020). Realized volatility forecasting with neural networks. Journal of Financial Econometrics, 18(3), 502-531.