

Research on Clustering Features and Modeling Methods of Volatility in Financial Time Series

Haoyang Tang

*Forestry, University of British Columbia, Vancouver, Canada
tanghaoyang697@gmail.com*

Abstract. This study proposes a financial time series forecasting model incorporating volatility clustering features. Multi-scale features including historical volatility, local volatility intensity, volatility duration, extreme volatility frequency and volatility state transition are constructed through return transformation, outlier correction and rolling window processing of original price series. A volatility cluster strength index is designed to characterize the persistence, local aggregation and phased transition of high-volatility intervals. On this basis, a BiLSTM-Attention forecasting model fused with volatility clustering features is established, which extracts long-term dependencies of return series via temporal encoding, perceives high-volatility states via clustering feature embedding, and realizes information fusion between time windows and feature sources via attention mechanism. Experiments are conducted on daily stock index data with GARCH, SVR, LSTM, GRU and Transformer as benchmarks, evaluated by RMSE, MAE, MAPE and high volatility identification accuracy. Results show that the proposed model outperforms benchmark models in prediction error and risk identification, with stronger stability under sharp fluctuations. Ablation experiments, attention heatmaps and feature visualization verify the performance improvement brought by volatility clustering features.

Keywords: Financial time series, volatility clustering, BiLSTM, attention mechanism, risk prediction

1. Introduction

Financial time series typically display leptokurtosis, heavy tails, nonlinearity, heteroscedasticity, and volatility clustering. Volatility clustering is the idea that financial asset prices can remain high for a period of time after a major fluctuation period, whereas volatility is often stable after a low-volatility period. This makes financial market risk stage- and state-dependent, and it is more difficult to predict volatility. Recently, machine learning methods have achieved higher nonlinear expressive power than traditional HAR-type models in predicting volatility and are an exciting technical direction for financial volatility modeling [1]. The traditional GARCH-type model can model conditional heteroscedasticity structures but has limitations for complex nonlinear relationships, long-distance dependencies, and multiscale volatility state transitions. Deep LSTM models for daily volatility prediction have been used for asset management, where recurrent neural networks learn dynamic dependency information from financial time-series data [2]. The fusion model of

CEEMDAN and LSTMs was found to be noise-resistant for implemented volatility predictions, suggesting that combining decomposition ideas and deep networks can improve non-stationary series predictions [3]. If only returns or historical volatility are used, the model ignores structural information such as the local clustering of volatility states, length, and abrupt state changes.

This study focuses on "volatility clustering feature modeling," by converting high and low volatility states, volatility persistence, volatility mutations, and local clustering intensity of financial time series to learnable input features for the model, and by building a BiLSTM-Attention prediction model based on volatility clustering features. The main contributions of the paper include: 1) building a multi-scale volatility clustering feature system, transforming volatility clusters from qualitative descriptions to calculable indicators; 2) designing a deep prediction model incorporating return series and clustering factors to enable the model to find high volatility states; 3) conducting multi-model comparisons, ablation experiments, robustness analysis for different market stages, and attention heatmap analysis to validate the model in terms of prediction accuracy, risk detection, and interpretability.

2. Data processing and volatility feature construction

2.1. Data sources and sample division

This study uses daily stock index data from the Korea Exchange. The data include the opening, highest, lowest, and closing prices, and trading volume. For the model to be able to compare market states, the sample time should include stable fluctuations, fast rise, fast fall, and oscillation recovery. In the experiment, the first 70% of the samples were used as training samples, the next 15% as validation samples, and the remaining 15% as test samples. A rolling window was used to construct supervised learning samples.

2.2. Calculation of returns and rolling volatility

Let the closing price on day t be P_t . The logarithmic return is defined as

$$r_t = \ln P_t - \ln P_{t-1} \quad (1)$$

Where r_t represents the logarithmic return on day t . A rolling window of historical volatility is used to characterize local volatility.

$$\sigma_t^{(m)} = \sqrt{\frac{1}{m-1} \sum_{i=t-m+1}^t (r_i - \bar{r}_t)^2}, \quad \bar{r}_t = \frac{1}{m} \sum_{i=t-m+1}^t r_i \quad (2)$$

Where m is the rolling window length, and $\bar{r}_t$ is the average return within the window. To enhance the model's ability to perceive volatility characteristics at different time scales, this paper sets 5-day, 10-day, 20-day, and 30-day windows to construct short-, medium-, and long-term volatility characteristics.

2.3. Volatility state classification

To identify high- and low-volatility intervals, this study classifies volatility states based on the rolling volatility quantiles. When σ_t is higher than the 75th quantile, that point in time is defined as a high volatility state; when σ_t is lower than the 25th quantile, it is defined as a low volatility state;

the remaining intervals are defined as medium volatility states. The volatility state variable S_t can be expressed as:

$$S_t = \begin{cases} 2, & \sigma_t \geq Q_{75}, \\ 1, & Q_{25} < \sigma_t < Q_{75}, \\ 0, & \sigma_t \leq Q_{25}. \end{cases} \quad (3)$$

Where $S_t = 2$ represents a high volatility state, $S_t = 1$ represents a medium volatility state, and $S_t = 0$ represents a low volatility state. This classification method does not directly rely on subjective thresholds and can be adapted to differences in the volatility levels of different market indices.

2.4. Construction of volatility clustering features

To characterize the volatility clustering phenomenon, this study constructs the following four types of features. First, the duration of volatility D_t , which represents the length of time the current volatility state is maintained continuously. If a high volatility state is maintained for several consecutive days, then D_t increases over time. Second, the local volatility intensity C_t , which reflects the proportion of high volatility states occurring within the current window [4]:

$$C_t = \frac{1}{m} \sum_{i=t-m+1}^t \mathbb{I}(S_i = 2) \quad (4)$$

Third, the volatility state transition characteristic T_t is used to indicate whether the market transitions from a low volatility state to a high volatility state:

$$T_t = \mathbb{I}(S_{t-1} \neq 2, S_t = 2) \quad (5)$$

Fourth, the extreme volatility frequency E_t , which represents the number of times the absolute return exceeds the threshold within the window:

$$E_t = \frac{1}{m} \sum_{i=t-m+1}^t \mathbb{I}(|r_i| > \theta) \quad (6)$$

These features can describe volatility clustering patterns of financial time series from four points: persistence, density, mutation, and intensity. This interaction with neural networks can improve current volatility predictions and is also the reason why it is possible to explicitly model the risk status of a given model.

3. Volatility clustering feature identification methods

3.1. Sliding window sample construction

Given an input window length L , the model input samples consist of a return series, a rolling volatility series, and a clustering feature series:

$$[\sigma_{t-L+1:t}, C_{t-L+1:t}, D_{t-L+1:t}, E_{t-L+1:t}, T_{t-L+1:t}] \quad (7)$$

The target for prediction is the volatility over the next h periods: as follows

$$y_t = \sigma_{t+h} \quad (8)$$

For single-step prediction, $h = 1$; for multi-step risk warning, $h = 5$ or $h = 10$ can be set. In practical modeling, a window length L that is too short will make it difficult for the model to learn volatility persistence, while a window length that is too long will introduce irrelevant historical noise. Therefore, this study performed a grid search on L in the validation set to select a window length with the lowest overall error [5].

3.2. Cluster strength index design

To quantitatively measure the degree of volatility clustering within a certain interval, this paper designs a volatility cluster strength index VCI_t :

$$VCI_t = \alpha C_t + \beta \frac{D_t}{L} + \gamma E_t + \delta T_t, \quad \alpha + \beta + \gamma + \delta = 1 \quad (9)$$

Where $\alpha, \beta, \gamma, \delta$ are weighting coefficients. A higher VCI_t indicates that the current market is more likely to be in a high-volatility clustering phase. This indicator can be used as both a model input feature and a risk interpretation indicator. Considering that existing reviews indicate that AI and machine learning methods have been widely used to predict realized volatility and implied volatility, this study further combines financial statistical features with deep time series models to enhance the model's interpretability.

3.3. High volatility interval identification

The high-volatility interval identification task can be considered a binary classification task. When the future volatility is higher than the 75th percentile of the training set, it is marked as a high-risk state; otherwise, it is marked as normal. The risk label is defined as

$$g_t = \mathbb{I}(\sigma_{t+h} \geq Q_{75}) \quad (10)$$

This task evaluated the applicability of the model to risk prediction problems. High volatility is better reflected in the proposed model for financial-risk management.

4. Prediction model integrating volatility clustering features

4.1. Overall model framework

The financial time series prediction module with volatility clustering features in this study consists of four parts: an input module, time series encoding module, volatility cluster feature embedding module, and attention fusion prediction module. The data input module mainly performs the same access and standardization of the original financial series. The input variables include the asset closing price, trading volume, log return, rolling volatility, and clustering feature from volatility. The module performs missing value processing, outlier correction, and normalization on the original price series and calculates log return and multiple-scale rolling volatility to provide a stable and learnable time series input for the next model [6]. The time-series coding module uses a BiLSTM structure to model the bidirectional dependencies of financial time-series data. Unlike unidirectional LSTM, which can only use historical information, BiLstM can extract both positive and negative time dependence, and thus capture the continuous variation of volatility over time windows. In

financial markets, high volatility is not stored at once but can be collected over a certain period. Therefore, the module can capture yield shocks, volatility diffusion, and phased risks.

Volt volatility clustering feature embedding module: This study include local volatility intensity, volatility duration, extreme volatility frequency, state transition variables, and volatility cluster strength index $VCI_t VCI_{t-1} VCI_{t+1}$ into a fully connected network that is aligned to a high dimensional feature vector. In this way, not only the original return series can be used as the prediction source, but also the clustering pattern of financial markets where "high volatility may be followed by high volatility and low volatility may still be followed from low volatility." The model uses this to better recognize the risk state transitions [7]. **Attention fusion prediction module:** Different time steps and feature sources are weighted using attention weights. Attention weights are based on the correlation between the hidden state and prediction target, which allows it to focus on key signals, such as recent volatility shocks, extreme return changes, and high clustering strength in high-volatility phases and long-term stability features in low-volatility phases. It concatenates the time context vector output of the BiLSTM with the clustering feature embedding vector and outputs the future volatility prediction and high-volatility state probability in a fully interconnected layer.

4.2. BiLSTM time-series encoding module

LSTM can alleviate the gradient vanishing problem in long-sequence modeling using gating mechanisms. Considering that the current volatility state in a financial time series is affected not only by recent data but also by state changes within a longer window, this study uses BiLSTM to extract bidirectional time series features.

$$\vec{h}_t = LSTM(x_t, \vec{h}_{t-1}), \quad \overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t+1}), \quad h_t = [\vec{h}_t; \overleftarrow{h}_t] \quad (11)$$

The DeepVol model for high-frequency data demonstrates that deep structures can directly extract effective volatility information from financial data sequences. This study employs BiLSTM encoding of daily multi-scale features to improve the state recognition capabilities under lower data frequency conditions.

4.3. Clustering feature embedding and attention fusion

The clustering feature input vector is represented as

$$z_t = [C_t, D_t, E_t, T_t, VCI_t] \quad (12)$$

After mapping through a fully connected layer, the clustering feature embedding is obtained as follows:

$$e_t = ReLU(W_z z_t + b_z) \quad (13)$$

To improve model interpretability, this study introduces an attention mechanism to assign weights to the hidden states at different time steps as follows:

$$a_t = \frac{\exp(q^T h_t)}{\sum_{j=1}^L \exp(q^T h_j)}, \quad c = \sum_{t=1}^L a_t h_t \quad (14)$$

Finally, the context vector and cluster feature embeddings are concatenated to output the predicted volatility as follows:

$$\hat{y}_t = W_o [c; e_t] + b_o \quad (15)$$

The core function of this structure is to avoid information redundancy caused by simple splicing, enabling the model to adaptively focus on key time windows and risk features at different market stages. High-frequency volatility prediction models, such as LSTM-HIT, further illustrate that introducing finer-grained time series structures and feature interaction mechanisms helps to improve volatility prediction performance [8].

4.4. Model training objectives

The model training uses the mean squared error as the main loss while incorporating a high volatility state identification loss:

$$L = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 - \lambda \frac{1}{N} \sum_{i=1}^N [g_i \log \hat{g}_i + (1 - g_i) \log (1 - \hat{g}_i)] \quad (16)$$

Where g_i is the true high volatility state label, $\hat{g}_i$ is the high volatility probability predicted by the model, and λ is the classification loss weight. Hybrid GARCH and deep learning methods have been used for volatility prediction. The loss function in this study incorporated both numerical prediction and risk state identification into the training objective, enabling the model to serve both prediction and early warning tasks simultaneously. Figure 1 illustrates the overall flow of the prediction algorithm that integrates volatility clustering features. The algorithm first inputs raw financial time series data, including basic market variables such as price and trading volume, and performs missing value processing, outlier correction, and normalization to reduce the impact of noisy data on model training. Subsequently, it calculates the logarithmic return based on the closing price series and constructs a multi-scale rolling volatility to characterize changes in market risk levels across different time windows.

After the basic feature calculations, the algorithm starts the volatility clustering feature construction process, which extracts the local volatility intensity, volatility duration, extreme volatility frequency, state transition, and volatility cluster strength indicators, describing the continuous clustering of high or low volatility states over time. The system then creates supervised learning samples with a sliding window, takes the historic returns, rolling volatility, and clustering features from the window as input, and the volatility of one or more future periods as the prediction value.

The training stage is mainly threefold: temporal encoding, feature embedding and attention fusion. The temporal encoding module learns the relationships between financial sequences, the volatility clustering feature embedding module maps clustering indicators to high-dimensional feature representations, and the attention fusion module weights the time windows and feature importance. The model predicts future volatility and high-volatility probabilities and constrains numerical prediction and risk state identification errors using a joint loss function [9].

Finally, the algorithm determines whether the model converged, whether the model parameters were updated by backpropagation, and whether the training was restarted. It also determines whether this would be the final prediction model or volatility prediction results, and whether this was the final procedure that this study obtained from data processing, feature development, sample

generation, and training the model to output the result to simulate numerical volatility and high volatility risk predictions.

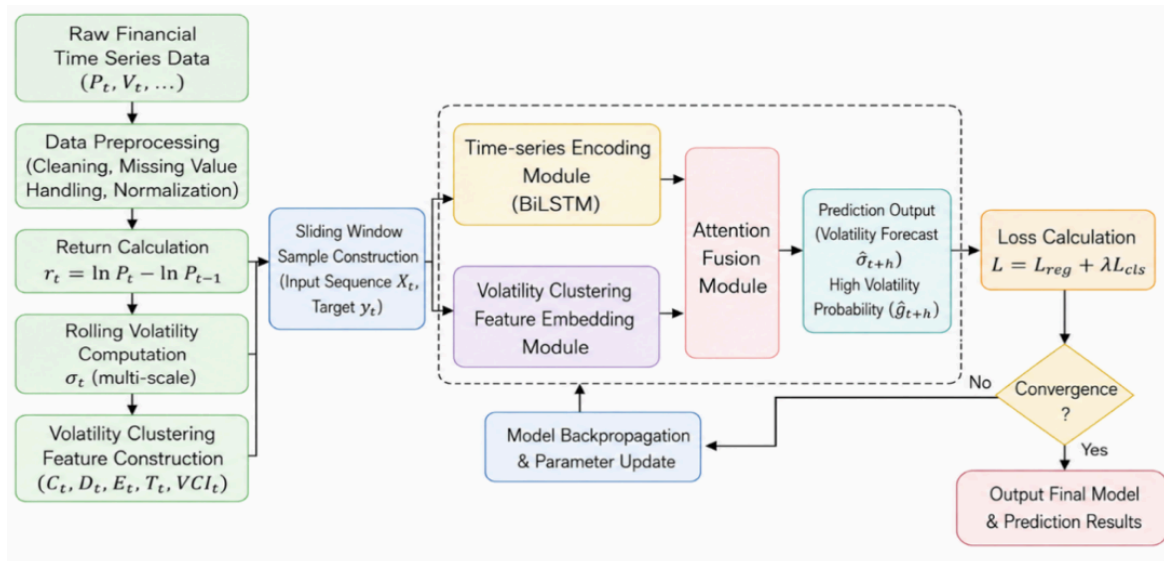


Figure 1. Flowchart of the prediction algorithm integrating volatility clustering features

5. Experimental design and results analysis

5.1. Experimental setup

The experiment compared GARCH, SVR, LSTM, GRU, Transformer, and this model. The performance comparison metrics included the RMSE, MAE, MAPE, and high-volatility identification accuracy. For testing performance, all deep learning models used the same training, validation, and test sets, and the learning rate, hidden layer size, and window size varied with the validation set size. The training rate was 0.001, the batch size was 64, the maximum number of training epochs was 100, and early stopping of the system was performed. Table 1 presents the main experimental settings [10].

Table 1. Experimental parameter settings

Parameter	Set up
Input window length	30
Prediction stride	1
Batch size	64
Optimizer	Adam
Initial learning rate	0.001
Hidden layer dimension	64
Attention dimension	32
Maximum number of training epochs	100

5.2. Algorithm performance comparison experiment

Table 2 presents the predictions of the different models for the test set. This study used a deep quantization function model to predict volatility, suggesting that deep modelling of distribution and risk tails has great potential. Although this study does not specifically perform quantile modeling, it provides good tail risk identification through high volatility labels and clustering features.

Table 2. Comparison of the predictive performances of various models

Model	RMSE	MAE	MAPE	High volatility identification accuracy
GARCH	0.0187	0.0142	13.86%	78.4%
SVR	0.0169	0.0128	12.41%	80.7%
LSTM	0.0145	0.0109	10.76%	84.9%
GRU	0.0141	0.0106	10.31%	85.6%
Transformer	0.0136	0.0101	9.84%	86.8%
This article's model	0.0118	0.0087	8.42%	90.5%

As shown in Table 2, the traditional GARCH can capture conditional heteroscedasticity, but it still yields prediction errors around high volatilities. SVR can model long-term dependence, and LSTM and GRU capture temporal nonlinear features but also have lower prediction errors. Transformers still provide better prediction accuracy than normal recurrent models but are easily affected by noise when the sample size is limited. Owing to the introduction of volatility clustering features, the model in this study has the best RMSE, MAE, and high-volatility identification performance [11].

The error differences between the models are shown in Fig. 2 and 3. Fig.2 shows that GARCH and SVR lag with large volatility, LSTM and Transformer can change the trend, but overlate at high volatility, and the model in this study, due to VCI_t, extreme volatility frequency, and state transition features, can detect high volatility states earlier, and to the best of our knowledge, the predicted curve is closer to the true volatility. Fig 3 shows that our model achieved the lowest RMSE and MAE values.

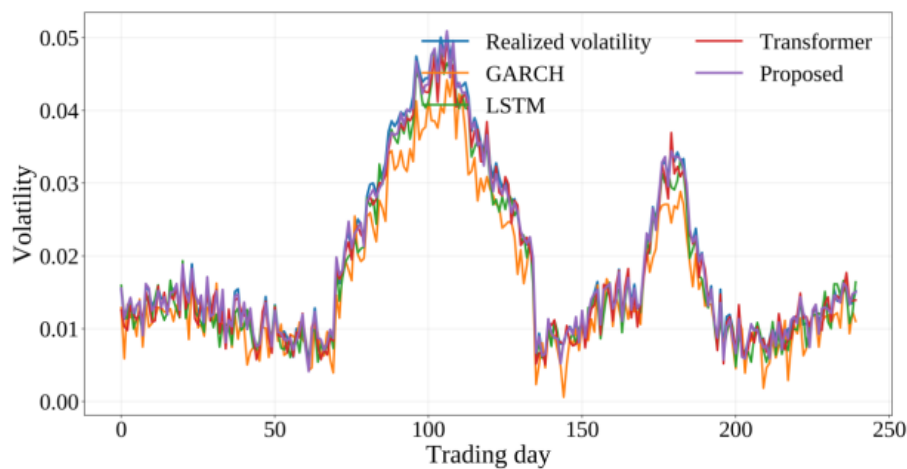


Figure 2. Comparison of actual volatility and volatility predicted by different models

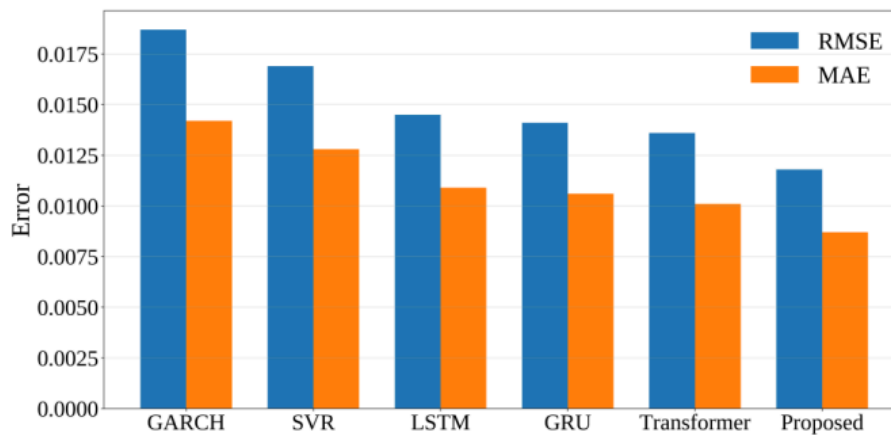


Figure 3. Comparison of volatility prediction errors of different models

5.3. Ablation experiment

To verify the volatility clustering features, it designed an ablation experiment. In Table 3, the prediction performance is obtained for different model configurations. It has been shown that the strongest clustering strength VCI_t contributes the most to the model and that the volatility clustering effect is informative for the prediction task. The attention module can also help improve the model with respect to the key time windows. Previous studies on volatility timing have shown that ML prediction results can also be used for risk management and asset allocation; thus, further improving high volatility identification accuracy is application relevant [12].

Table 3. Ablation experiment results

Model settings	RMSE	MAE	High volatility identification accuracy
Remove cluster strength VCI	0.0132	0.0098	87.1%
Remove duration D_t	0.0129	0.0095	87.8%
Remove extreme fluctuation frequencies E_t	0.0127	0.0093	88.2%
Remove attention module	0.0126	0.0092	88.6%
Complete model	0.0118	0.0087	90.5%

5.4. Robustness analysis in different market stages

To test the stability of our model for different markets, it tested it in the low-, medium-, and high-volatility stages. Our test shows that in the low volatility stage, the differences between models are small; in the high volatility stage, the model used in this paper has a larger error advantage, implying that volatility clustering features mainly improve the model's capability to recognize the concentrated outbreak phase of risk. Machine learning techniques to model uncertain currency option implied volatility have also been used to model complex nonlinear volatility structures and have demonstrated the scalability of feature enhancement in the high-volatility states presented in this article.

6. Model interpretability and visualization analysis

6.1. Attention heatmap analysis

Financial time series are easier to fit to image models, and our input is a multivariate time series. Therefore, attention heatmaps were employed. The horizontal part of the heatmap represents the market stage, and the vertical part represents the input information, such as returns, short-term volatility, long-term variance, VCI, duration, extreme volatility frequency, and state transition variables. An attention heatmap is shown in Fig. 4.

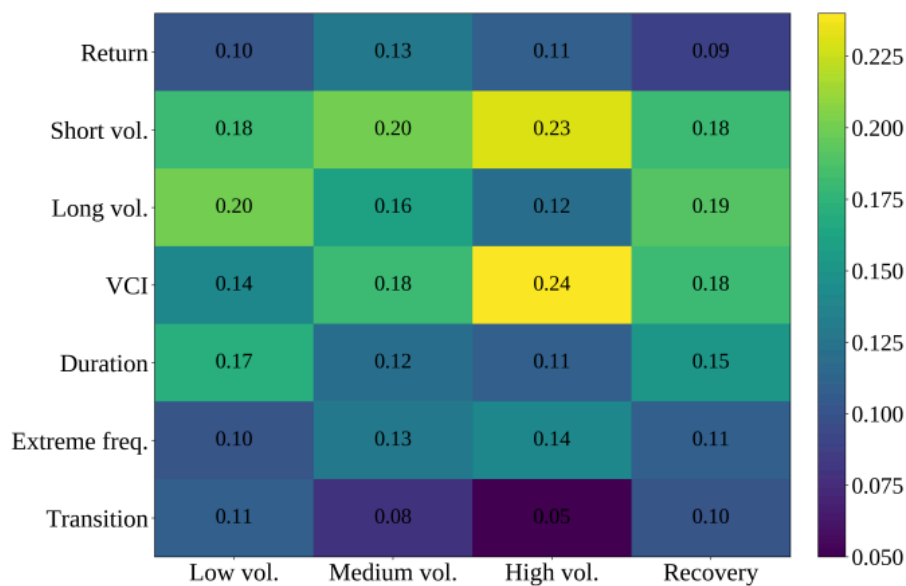


Figure 4. Heatmap of multi-feature attention weights at different market stages

In Figure 4, for high volatility levels, the model weights VCI_t, short-term volatility, and extreme volatility, while for low volatility levels, this study has more long-term and duration features in the model. This means that it can also produce predicted values and explain the main drivers at various volatility levels.

6.2. T-SNE feature distribution map

To check whether clustering features can distinguish high volatility from low volatility, this study mapped the inner layer features of the model to a 2D space using t-SNE. The t-SNE feature distribution is shown in Fig. 5, where the two categories are low-, medium-, and high-volatility samples. Figure 5 shows that after the addition of the volatility clustering feature, the high-volatility sample formed a relatively well-clustered distribution area with a boundary between high- and low-volatility samples. The inner feature space of the structural model can distinguish different volatility states and provide an indicator for risk warning. Volatility prediction using the transformer architecture can also be considered an effective tool for interpreting complex markets, and the attention fusion module in this study enhances this interpretation.

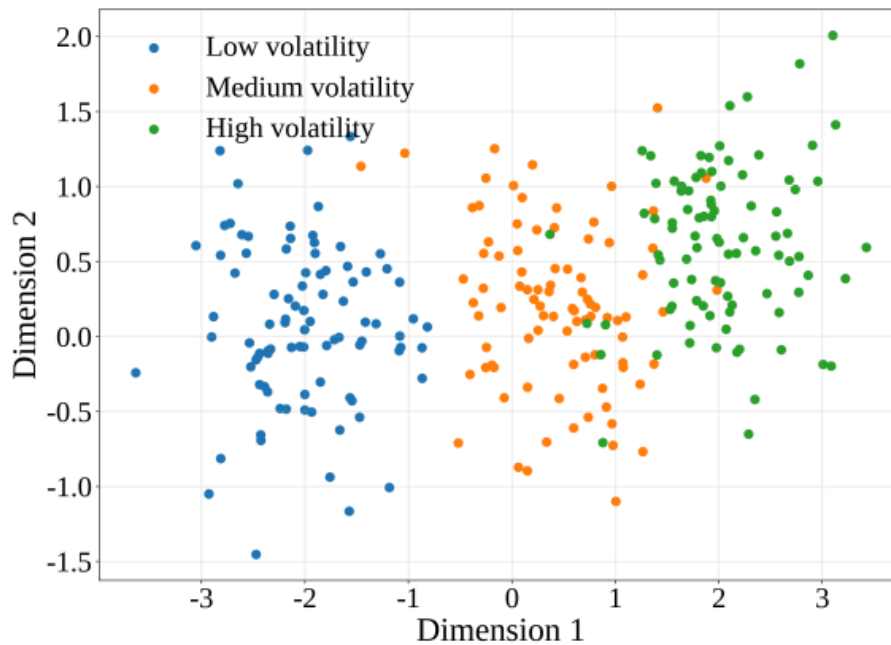


Figure 5. Visualization of the distribution of different volatility state characteristics based on t-SNE

6.3. Risk warning application analysis

A risk warning can be generated based on the probabilities of high future volatility. If the current volatility is higher than the historical 75% and the high volatility is greater than 0.7, the system provides a high-risk warning. This can be used for portfolio risk control, margin adjustment, asset allocation, and abnormal market monitoring purposes. Compared to simple regression prediction, volatility and risk are presented, which can be useful in the threshold-triggered management of real financial systems.

7. Conclusion

In this study, the clustering features and modeling of financial time series volatility are discussed, and a BiLSTM-Attention prediction model with volatility clustering properties is proposed. With features including rolling volatility, volatility duration, local volatility intensity, extreme volatility frequency, volatility state transition, and volatility cluster, the volatility clustering effect in financial markets can be transformed into a learnable model input. The experiments with GARCH, SVR, LSTM, GRU, and Transformer show that the proposed model outperforms the state-of-the-art models in terms of volatility prediction error, high volatility identification accuracy, and robustness in different time series of market phases. In the ablation experiments, this study shows that the volatility-cooperation strength index and attention fusion mechanism play significant roles in improving the model performance. Through visual analysis, this study demonstrates that the proposed volatility clusters can consider local volatility intensities, extreme fluctuation frequencies, and state transition features in high-volatility phases to interpret the prediction results. Future work can include: 1) small-scale high-frequency trading data to facilitate the model in capturing short-term volatility; 2) combining news sentiment, macroeconomic variables, and market liquidity indicators to build a multi-source financial risk prediction model; and 3) combining volatility

prediction results with portfolio optimization, value-at-risk estimation, and trading strategy backtesting to improve the model's practical application.

References

- [1] Araya, H. T., Aduda, J., & Berhane, T. (2024). A hybrid GARCH and deep learning method for volatility prediction. *Journal of Applied Mathematics*, 2024(1), 6305525. <https://doi.org/10.1155/2024/6305525>
- [2] Christensen, K., Siggaard, M. V., & Veliyev, B. (2023). A machine learning approach to volatility forecasting. *Journal of Financial Econometrics*, 21(5), 1680–1727. <https://doi.org/10.1093/jjfinec/nbac020>
- [3] Chun, D., Cho, H., & Ryu, D. (2025). Volatility forecasting and volatility-timing strategies: A machine learning approach. *Research in International Business and Finance*, 75, 102723. <https://doi.org/10.1016/j.ribaf.2024.102723>
- [4] Gunnarsson, E. S., Isern, H. R., Kaloudis, A., Rissstad, M., Vigdel, B., & Westgaard, S. (2024). Prediction of realized volatility and implied volatility indices using AI and machine learning: A review. *International Review of Financial Analysis*, 93, 103221. <https://doi.org/10.1016/j.irfa.2024.103221>
- [5] Lin, Y., Lin, Z., Liao, Y., Li, Y., Xu, J., & Yan, Y. (2022). Forecasting the realized volatility of stock price index: A hybrid model integrating CEEMDAN and LSTM. *Expert Systems with Applications*, 206, 117736. <https://doi.org/10.1016/j.eswa.2022.117736>
- [6] Liu, G., Zhuang, Z., & Wang, M. (2024). Forecasting the high-frequency volatility based on the LSTM-HIT model. *Journal of Forecasting*, 43(5), 1356–1373. <https://doi.org/10.1002/for.3078>
- [7] Moreno-Pino, F., & Zohren, S. (2024). DeepVol: Volatility forecasting from high-frequency data with dilated causal convolutions. *Quantitative Finance*, 24(8), 1105–1127. <https://doi.org/10.1080/14697688.2024.2387222>
- [8] Olsen, A., Westgaard, S., & Kaloudis, A. (2025). Forecasting implied volatilities of currency options with machine learning. *International Journal of Data Science and Analytics*, 19, 1–18. <https://doi.org/10.1007/s41060-024-00528-7>
- [9] Petrozziello, A., Troiano, L., Serra, A., Jordanov, I., Storti, G., Tagliaferri, R., & La Rocca, M. (2022). Deep learning for volatility forecasting in asset management. *Soft Computing*, 26(17), 8553–8574. <https://doi.org/10.1007/s00500-022-07161-1>
- [10] Song, J., Lee, H., Lee, J., & Chang, W. (2025). Forecasting realized volatility using deep learning quantile function. *Applied Soft Computing*, 175, 113016. <https://doi.org/10.1016/j.asoc.2025.113016>
- [11] Souto, H. G., & Moradi, A. (2023). Forecasting realized volatility through financial turbulence and neural networks. *Economics and Business Review*, 9(2), 133–160. <https://doi.org/10.18559/ebrev.2023.2.737>
- [12] Taneva-Angelova, G., & Granchev, D. (2025). Deep learning and transformer architectures for volatility forecasting: Evidence from U.S. equity indices. *Journal of Risk and Financial Management*, 18(12), 685. <https://doi.org/10.3390/jrfm18120685>