Implied Volatility Prediction of Financial Options Products Based on *the CL-TCN Model*

Yuemeng Li, Chenyu Wang, Zhongchen Miao, Jian Gao, and Jidong Lu(✉)

Innovation Lab, Shanghai Financial Futures Information Technology Co., Ltd, Shanghai, China
lujd@cffex.com.cn

Abstract. The implied volatility of options is a key factor in judging the price trend of options and analyzing their trade, so it is very important to use a reasonable method to predict it accurately. Since the traditional B-S-M formula calculation method cannot reflect the actual changes of Tick-level granularity implied volatility in the market, we found a model suitable for processing option quotation data with significant high-frequency and fine-grained timing features, which named CL-TCN model. It combines the contrastive learning framework and TCN model, and its special time series encoding method to predict the downstream implied volatility task is of great help, which can not only solve the problem of order discontinuity caused by the difference in option liquidity, improve the parallel processing efficiency of high-frequency time series data, but also improve the accuracy and generalization of forecasting. At the same time, this paper also mines the features of option-related business, verifies the endogeneity of the model by using clustering algorithm, and extracts a more representative volatility analysis indicators than the traditional calculated variables.

Keywords: Implied Volatility · Contrastive Learning · TCN · Clustering · Feature Extraction

1 Introduction

In the pricing model of options, volatility appears as a very critical element, which is both a term for the price changing frequency and an implicit feature of the price movement; Some of those who trade options are more experienced in dealing with volatility risk than price risk. When volatility is calculated using past prices, it is called historical volatility, and it is usually benchmarked against the speed of past price changes. The volatility of the underlying in the options market during the life of the option is called implied volatility, which usually indicates the market's expectation for a period in the future. Therefore, the implied volatility represents the market trader's risk appetite for that option. When underlying assets become riskier in the future, the estimation of implied volatility will increase, which raises the pricing of options.

The implied volatility of the same option contract tends to vary at different times. Some factors can be predicted, such as the closer to the delivery date and the flat value, the greater the liquidity, which can correspond to higher or lower volatility. While some

J. Yen et al. (Eds.): ICBIS 2023, AHCS 14, pp. 572–587, 2023.
https://doi.org/10.2991/978-94-6463-198-2_59

can not be predicted, such as a company entering into a more speculative business area, the volatility will be increase. A test based on the predictive power of implied volatility is a joint test of whether the market is efficient and whether the option pricing formula is correct. For the formula method, if the actual price of an option is a reasonable price, it can be used as a fixed value of the B-S-M formula, and let the volatility become an unknown variable, which can be iteratively calculated. But when the power of implied volatility prediction does not cover historical volatility, in addition to the perception that the market is invalid, and people's judgment is inaccurate, there is also a possibility that it is inappropriate to use the option pricing formula to calculate implied volatility. Therefore, the prediction of implied volatility with the help of predictive models becomes even more important.

At the same time, in the option trading scenario, the option product has a large series, and each series corresponds to a different exercise price. Thus, the large amount of high-frequency, fine-grained information contains more detailed investment signals than the original granular data.

For the use of fine-grained information, it may be more difficult to predict the business data with natural discontinuity, such as it may have disturbance problems like empty order price or zero trading lots, or unbalanced time series intervals. The contrastive learning technique based on self-supervised representation learning can project features highly correlated with volatility into a high-dimensional feature space to better represent the inherent structure of the time series. At the same time, using adjacent and synchronously long time slices across time dimensions as the objects of comparative learning, can be avoided the dependence on a single time point. Besides, by constructing a specific loss function to train the encoder to obtain the difference in representation of positive and negative samples, the method can make the same subsequence have consistent expression in two different enhanced context views without introducing an inductive bias.

For the effective use of high-frequency information, in this paper, we also use the image-based convolutional neural network model variant - time convolutional network (TCN), which can learn a wider range of time series information and greatly improve the efficiency of time series embedding generation.

In summary, compared with the original method, our model has been improved and innovated in the following three aspects:

- Obtain the time series data of each product arranged daily from morning to evening in the past 45 days and introduce the method of contrastive learning to distinguish positive and negative samples in the option time dimension, which enhances the inherent structure of time series features, and can capture multi-distribution and multi-scale context information because of reducing labeled data.
- Using the TCN model, the operation of multi-layer pooling and expansion convolution is used to obtain a time series representation with a larger receptive field, that is, it can take into account the long-term time series information, improve the training efficiency and obtain the same excellent performance as RNN in the time series model, and increase the practicability of the high-frequency time series encoding model in real scenarios.
- In business, different business characteristics are introduced for training, and the characteristics of volatility are better characterized by the model fitting the volatility.

Then, the feature code is extracted for cluster analysis, and the relationship between the cluster pattern and the expiration time, execution price, and volatility are analyzed, and more business value is mined.

2 Related Work

The time series analysis task for the purpose of predicting the implied volatility of options is a relatively novel study. Traditional time series forecasting models, such as ARIMA and GARCH, mainly rely on mining the statistical distribution law of historical volatility to build models related to their own past based on autoregressive theory (usually the predicted results are relatively close to the historical average).

The well-known flaw of RNN networks is that it is easy to forget the first input, and LSTM effectively solves this problem by using the method of long and short memory, so LSTM [1] has a wider application. Methods such as LSTM, GRU [2], and others imply the assumption that the present state of the layer depends on the state of the previous moment. The emergence of various variant networks (such as ALSTM [3]) thus combines the local layer states at several points in time, so that not only the previous state can be seen, but also the historical state farther before can be seen, but the flaws of the network structure itself are still not solved. Shiyang Li, et al. NIPS 2019 [4] Transformer can model both long-term dependence and short-term dependence to see the current forecast distribution of attention to historical values, where Multi-head Attention can also focus on different patterns, while eliminating gradient explosions compared to RNNs, resulting in faster parallel computation. The emergence of TCN [5] also provides the possibility of massively parallel computing of time series data, making one-dimensional convolutional deformation suitable for timing problems, and in order to improve accuracy, it also adds a hopping layer connection of residual convolution and a 1×1 convolution operation.

On the other hand, due to the fact that there are many types of options product contracts and time series data cannot be sampled regularly, it is necessary for us to investigate the time series prediction scheme of non-regular sampling. One basic way to handle irregular sampling is fixed time discretization. For example, dispersing observations of continuous time into boxes up to an hour long, while simple, requires special handling of boxes with multiple observations and allows for data loss when the boxes are empty. An alternative to time discretization is to build a model that can directly use irregularly sampled time series as input. Che et al. (2018) [6] proposes several methods based on gated cyclic cell networks, which takes as inputs a sequence consisting of observations, missing data indicators, and time intervals since the last observation. Pham et al. (2017) [7] proposes to capture time irregularities by modifying the LSTM's forgetting gate, while Neil et al. (2016) [8] introduces a new time gate to regulate access to the hidden unit of the LSTM. While these methods allow networks to process vector values with irregular intervals, some recent methods also use attention mechanisms to simulate irregular sampling time series (Song et al., 2018 [9]; Tan et al., 2020 [10]) However, due to the lack of alignment of observation times between different product contracts, these methods do not support learning directly from partially observed samples.

In addition, recent research results (Eldele et al. 2021 [11]; Franceschi, 2019 [12], using contrast losses, reduces the requirement for observational time alignment from the inherent structure of learning time series, using random subsequences from the original time series as positive samples, but failing to distinguish between multi-scale contextual information, which is critical to the success of time series tasks. The Ts2vec [13] model solves this problem by improving the generalization ability of learning representations through different levels of information provided by multi-scale features.

3 Methodology

3.1 Feature Selection

In order to better illustrate the problem, we are trying to solve, it is necessary to fully explain the terminology widely used in the risk and price assessment of the options market. In this article, we have selected the following types of features that are frequently used in theoretical and practical business data analysis and are highly correlated with options. It can be divided into two main categories: the B-S-M formula is necessary to calculate the volatility; A commonly used implied volatility assessment metric for your business.

The variables required for B-S-M formula calculation are:

- **Option expiration date:** refers to the last date that the option contract must perform, calculated according to the trading calendar.
- **Risk-free interest rate:** refers to the interest rate that can be obtained by investing funds in an investment object without any risk, usually a fixed constant for a period.
- **Underlying price:** can correspond to the futures or spot price, the futures price refers to the futures price involved in the CSI 300 stock index option contract, under the condition that the option is finalized and the price is fixed, the level of the option price is largely determined by the futures price. The spot price refers to the contract price reached by the two parties to the transaction of the actual goods in accordance with the principle of fairness, which in this case refers to the price of the spot stock of the CSI 300, because the futures price of the stock index is always based on the spot price of its own subject matter.
- **Current price:** the fee paid by the buyer and seller of the option to the option seller when the option buyer buys the option through the auction, that is, the premium of the option contract.
- **Strike price:** The strike price of the option, also known as the option agreement price or the option exercise price, refers to the price at which both parties to the option transaction agree to execute the call and put contracts within a specified period in the future.

In the business, different features are introduced to represent and predict downstream characteristics, and the characteristics that better characterize the volatility are better characterized by the model fitting the volatility. Common evaluation indicators are:

- **Remaining period:** The calculation method is *"maturitydate − tradingday"* and the implied volatility has a strong correlation with the remaining period.
- **Value state:** $ln(exerciseprice/underlyingprice)/\sqrt{res\_date}$, according to the "bias" and "curvature" of the value state, judge the change trend of value over time.

- **Exercise price *spacing/index***: According to the exercise price range, judge the "*spacing value/exercise price*", and control the change within a certain range under the principle of pending, which objectively affects the volatility.
- **Logarithmic yield:** $log(settlementprice/openingprice)$, a more noteworthy way of expressing the additive yield, which is potentially related to volatility.
- **Bid-ask spread, bid-ask price ratio:** the general market data only provides the market 1 file of trading information, in order to more fully understand the market entrusted market information, you also need to understand the depth of five levels of market can better portray the market trend:

  a. *ask1, ask2,..., ask5*
  b. *bid1, bid2,..., bid5*
  c. *ask1-bid1, ask2-bid2,..., ask5-bid5*
  d. *ask1/bid1, ask2/bid2,...,ask5/bid5*

The above features provide more angles to portray the value of the option itself, especially for the commonly used evaluation indicators in the business, which are parameters other than the pricing model, if validated and effective, can be taken into account when predicting the price trend of the option, and can form the judgment and decision of the volatility trend faster.

After we train the feature coding of these features through the encoder, in addition to extracting the code to make downstream predictions, we can also reverse verify the quality of the code, for example, the extraction of the feature code can be used for cluster analysis, analyzing the relationship between the cluster mode and the product, expiration time, execution price, and volatility to mine more business-valuable conclusions.

3.2 Contrastive Learning Module Designs

Contrastive learning is a promising class of self-supervised representational learning methods. Self-supervised learning hopes to learn an encoder, leverage the semantic dependencies of the data to build training goals, and obtain a common feature for the input representation of the underlying task. Specifically, for a bunch of unsupervised data entered, through the structure or characteristics of the data itself, artificially construct labels, that is, learn an embedded function:

$$v = f_\theta(x) \tag{1}$$

It is a deep neural network with a parameter θ that maps the time series x to feature v, which can obtain a feature representation of the essence of the data and improve the efficiency of data utilization. In this experiment, we use the two-dimensional data composed of time and feature dimensions to sample adjacent, synchronized long time slices in the time dimension as objects for comparative learning, so that a measure can be generated on the high-dimensional feature space, for example, to obtain enhanced representations of the time series context, such as the slice subsequences x and y, can be constructed:

$$d_\theta(x, y) = ||f_\theta(x) - f_\theta(y)|| \tag{2}$$

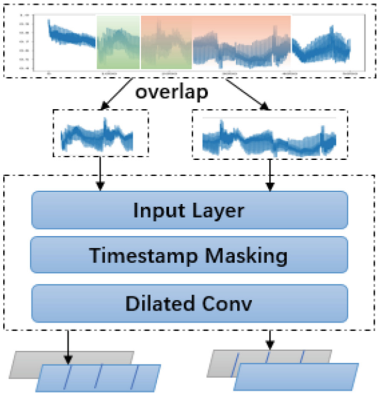


Fig. 1. Schematic diagram of comparative learning of time slice vectors

and by training the embedding so that the two vectors are closer together, the above values are smaller. The details are shown in the following (Fig. 1):

The usual loss function formula used in contrast learning is info NCE loss, which is used in multi-classification scenarios:

$$L_q = -\log \frac{\exp(q \cdot k_+/\tau)}{\sum_{i=0}^{k} \exp(q.k_i/\tau)} \tag{3}$$

However, in our current time series prediction scenario, in order to learn the difference representation of time changes, it is necessary to take the representation of the same timestamp as positive and the representation of different timestamps as negative when the input two-time subseries is entered. Let i be the index of the input timeseries sample, t is the timestamp, then $r_{i,t}$ and $r'_{i,t}$ means the same timestamp t but twice enhanced for x_i. In this way, the time contrast loss of the ith time series at the timestamp t can be expressed as:

$$l_{temp}^{(i,t)} = -\log \frac{\exp(r_{i,t} \cdot r'_{i,t})}{\sum_{t' \in \Omega} (\exp(r_{i,t} \cdot r'_{i,t'}) + 1_{[t \neq t']} \exp(r_{i,t} \cdot r_{i,t'}))} \tag{4}$$

3.3 Feature Extraction Module Design

Time series forecasting has been a very hot research topic in recent years. For a short period of time, RNNs' innate cyclic autoregressive structure makes it a good representation of time series, including LSTMs (long short term memory) and GRUs (gated recurring units) or improved ALSTMs. Because traditional convolutional neural networks are generally considered to be less suitable for modeling time series problems, this is mainly due to the limitation of the size of their convolutional kernels, which cannot grasp long-term dependent information well. Therefore, we use the convolutional neural network model variant - time convolutional network (TCN), which can learn a wider range of time series information by increasing the network depth, expanding the

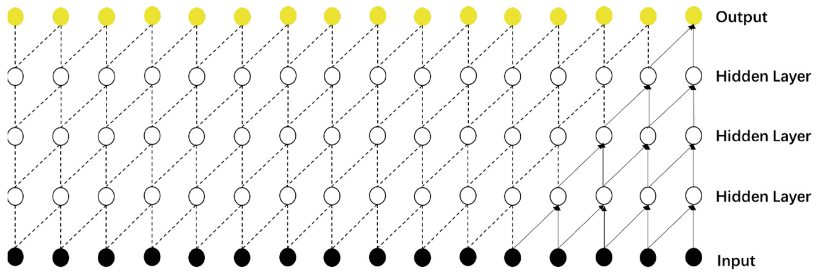


Fig. 2. Schematic diagram of causal convolution

receptive field and using the residual network, and at the same time, large-scale parallel computing also greatly improves the efficiency of time series model training, and the process is faster than RNN training, making self-supervised training of high-frequency market data possible.

It mainly contains three aspects of convolutional network characteristics, causal convolution, expansive convolution, and residual connections.

3.3.1 Causal Convolution

For the value of the t-moment of the previous layer, only the value of the next layer of t-moment and before it is dependent. The difference with the traditional convolutional neural network is that causal convolution does not see the future data, it is only a one-way structure, only the previous cause can derive the subsequent effect, is a strict time-constraint model, and can achieve parallel processing, convenient for the self-supervised learning process of large-scale data (Fig. 2).

3.3.2 Bulking Convolution

Purely causal convolution or the problem of traditional convolutional networks, that is, the modeling length of time is limited by the size of the convolutional kernel, and if you want to grasp a longer dependency, you need multiple layers of linear stacking, so the essence of TCN is the use of expansion (hole) convolution, as follows:

Expansive convolution allows the input to be sampled at intervals when convoluting, and the sampling rate is controlled by d in the plot. The d = 1 at the bottom level means that each point is sampled when entering; The middle layer d = 2 means that every 2 points are sampled as input when entering, and the higher the level the larger the size of d used. So, the expansion convolution causes the size of the effective window to grow exponentially with the number of layers. In this way, convolutional networks can obtain a large sense of field with fewer layers (Fig. 3). At the same time, the sensing field can be flexibly defined, determined by the number of layers, the size of the convolutional nucleus, and the expansion coefficient. The formula looks like this:

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot \mathrm{x}_{s-d\cdot \mathrm{i}} \tag{5}$$

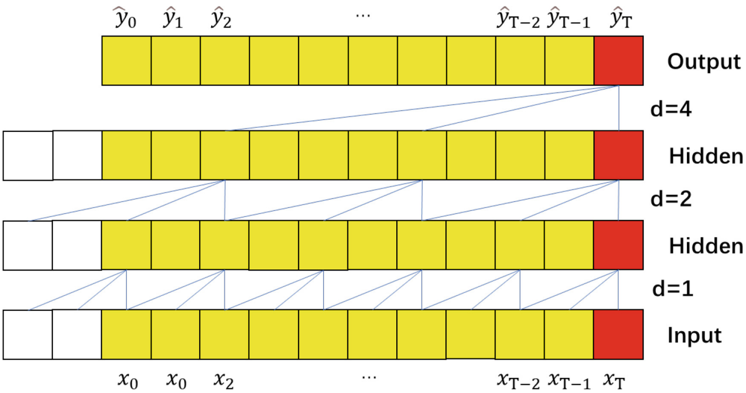


Fig. 3. Schematic diagram of Bulking convolution

3.3.3 Residual Links

It can train deep networks and make them transmit information across layers, so it can build a residual block instead of a layer of convolution and add regularization weights and dropout to regularize the network, solving the problem of gradient disappearance and explosion that often occurs in RNN.

$$o = Activation(x + F(x)) \tag{6}$$

3.4 Full Figure of Constractive-Learning-TCN(CL-TCN)

The above (Fig. 4) shows the entire network structure. The vector pairs trained by comparative learning are fed into the TCN network for parallel convolution, pooling, and residual operations. The result is in a high-dimensional space is:

$$\{z_1, z_2, \ldots, z_{t-1}, z_t, \ldots, z_T\} \Rightarrow \{z_1{\prime}, z_2{\prime}, \ldots, z_{t-1}{\prime}, z_t{\prime}, \ldots, z_T{\prime}\} \tag{7}$$

4 Experiment

4.1 Data Selection, Sampling and Processing

In this paper, roughly 31,180 sample points are sampled as experimental research objects in the adjacent parity options of the CSI 300 stock index options that are close to the delivery date and the 3-strike price spacing that floats up and down the parity option and float up and down the parity option according to the time granularity of 500 ms. The reason for this selection is that the closer to the delivery month, the larger the amount of options, the greater the liquidity. At the same time, due to its larger price fluctuation space, balanced investment opportunities, and moderate risk, the flat value contract is also the most attractive for investors to trade, and our analysis of it is more meaningful.

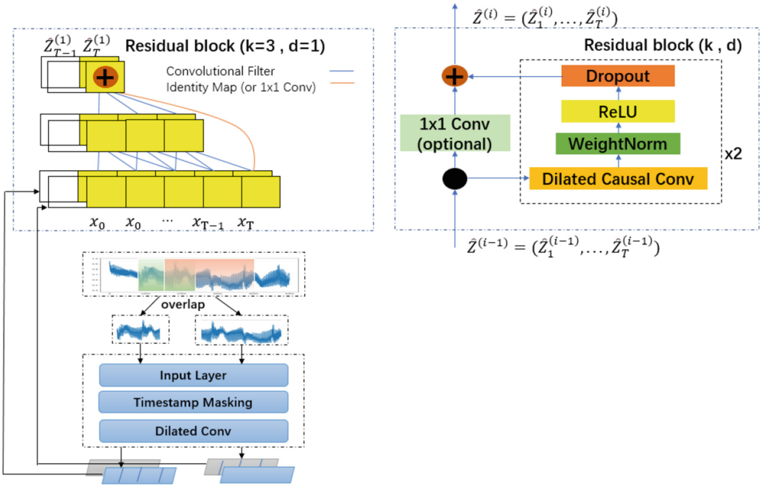


Fig. 4. Schematic diagram of CL-TCN

The time granularity of 500 ms was chosen not only because the option contract generates a price order at 500 ms fixedly, but also because when using more frequent fine-grained data, it can carry input signals that exceed the coarse granularity by tens or hundreds of times, such as tick and minute level data that contain much more information than the daily trend label (32-bit return), making it easier to provide direct feedback signals to obtain time-sensitive implied volatility trend predictions.

In terms of sampling method, since the option contract contains more than 200 types, the trading interval of each same contract is not exactly in accordance with 500 ms, resulting in the generation of non-equal interval data weakening the inherent timing characteristics and coherence. At the same time, the contracts traded every day are not exactly the same, so it is necessary to make effective trading interval filling and sample equilibrium of different contract products to objectively reflect the law of volatility. Since we introduced a contrast-learning model for training, we need to construct specific positive and negative sample pairs suitable for contrastive learning. Our construction method is to use synchronized long-time slices as objects for comparative learning, as shown in the following (Fig. 5):

For adjacent time slices with overlapping parts, they can be regarded as positive sample pairs; If there is no overlap between neighbors, it is a negative sample pair.

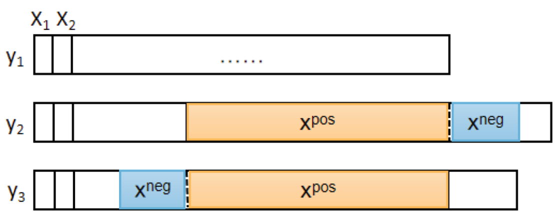


Fig. 5. Compare the positive and negative sample representation graphs

4.2 Design of Experiments

This experiment tests the model effect from two aspects.

(1) Evaluates the exogenous nature: There is the downstream result verification, which evaluates the exogenous nature of the model by predicting the line chart trend of the result (Fig. 6) and comparing it with the actual line chart trend (Fig. 7).

We selected the sampled data over time to plot the true waveform of the implied volatility. The waveforms generated by the model predictions were compared. It is found that when the embedding generated by CL-TCN is used by LR-model for downstream prediction, when the fine-grained time interval reaches about 50,000 points, that is, the amount of training data is 50,000 points, the generated prediction waveform can roughly reflect the real trend. LSTMs never seem to reflect a similar effect.

At the same time, we use 5000 points of randomly sampled data as the prediction set, evaluate the MSE and MAE values of the following models respectively (Table 1), and plot their predicted waveforms and real waveforms for comparison (Fig. 8).

From the above, it can be seen that the CL-TCN model is better than other models. We evaluate the training efficiency of various models and find that CL-TCN has improved the training speed compared with other time series models.

(2) Evaluate the endogeneity: We selected two feature combination methods, compared the quality of intermediate high-dimensional features generated by CL (contrastive-learning) and CL-TCN, and verified the effect by clustering (Table 2).

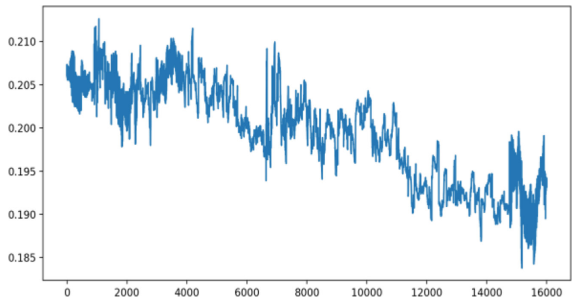


Fig. 6. Figure of Ground-Truth

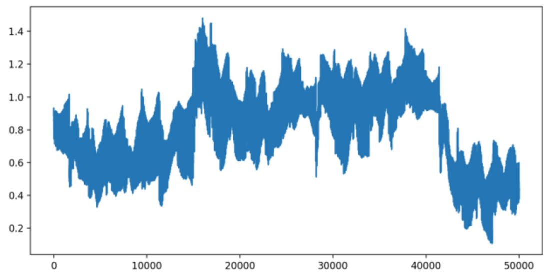


Fig. 7. Figure of CL-TCN + LR (50000 points)

Table 1. How different models predict the best combination of features.

The Model of Prediction	MSE	MAE	The Time to training (N = 40000, batch = 32)/Epoch
LSTM(b)	0.301	0.273	62 s
Attention-based(c)	0.344	0.587	37 s
CL-TCN + LSTM(d)	0.188	0.137	4.5s(only CL-TCN)
CL-TCN + Attention-based(e)	0.194	0.1395	Ibid.
CL-TCN + ATT-LSTM(f)	0.193	0.1392	Ibid.

Among them, the clustering mode of CL-TCN can better present some internal conclusions related to feature selection, so as to verify the effect of CL-TCN model and evaluate the endogeneity of the model:

a. For the first set of features **<years of time, bid-ask spread, bid-ask price ratio>**, which have a stronger correlation with the annualized expiration time, so we study the term variables, select a batch of data with the same execution price in the current month, and perform PCA dimensionality reduction and k-means clustering on the CL-encoded vector, and obtain the following two clustering results as Fig. 9.

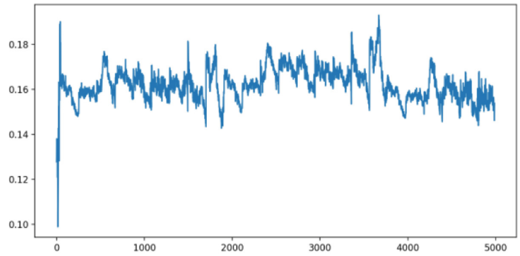
Analyzing the sample characteristics of each category, it is found that the current time included in the first category is the four dates of the 1^{st}, 2^{nd}, 8^{th} and 9^{th} of the month, and the current time included in the second category is the 10^{th}, 13^{th} and 14^{th} of the month.

b. Similarly, for the first set of features **<years of time, bid-ask spread, bid-ask price ratio>**, a batch of data with the same execution price in the current month was selected, and PCA dimensionality reduction and k-means clustering were performed on the CL-TCN encoded vector, and the following four clusters of clustering results were obtained as Fig. 10.

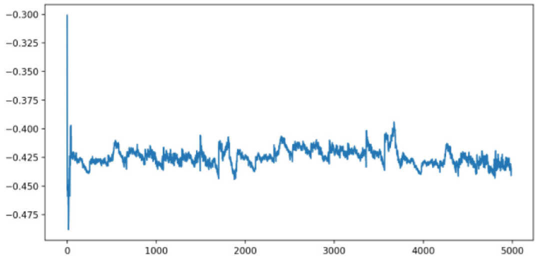
Analyzing the sample characteristics of each category, it is found that the first and third categories mainly contain the two dates of the 13^{th} and 14^{th} of the month; The second category mainly contains two dates: the 1^{st} and 2^{nd} of the month; The fourth category mainly contains three dates: the 8^{th}, 9^{th} and 10^{th}. And from the effect point of view, the distinction between features is stronger.

c. For the second set of features <moneyness, bid-ask spread, bid-ask price ratio>, which have a stronger correlation with the value state of the option, so we study the value variable, select a batch of data with the same expiration date, but the strike price is above and below the flat value 3 exercise price spacing (in units of 50), and perform PCA dimensionality reduction and k-means clustering on the CL-encoded vector, and obtain the following 5 Cluster clustering results as Fig. 11.

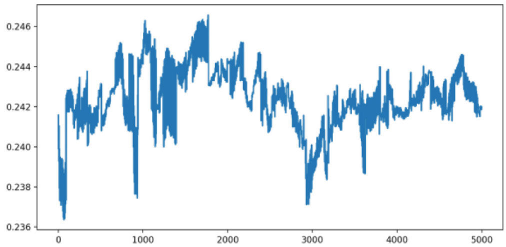
d. Similarly, for the second set of features <moneyness, bid-ask spread, bid-ask price ratio>, a batch of data with the same expiration date but 3 exercise price spacings (in units of 50) above and below the strike price was selected, and PCA dimensionality reduction and k-means clustering were performed on the CL-TCN encoded vector, and the following five clusters of clustering results were obtained as Fig. 12.



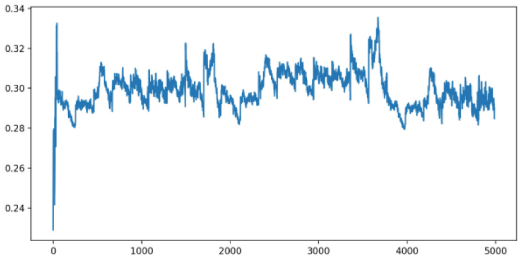
(a) Ground-truth



(b) LSTM



(c) Attention-based



(d-f) the models of CL-TCN

Fig. 8. Group plot of prediction results

Analytical experiments *c* and *d*, the spot target price of this batch of samples is 4050, and analyzing the sample characteristics of each type, it is found that the execution price of the first type is 3900 and 3950, the execution price of the second and third types is 4000, 4050 and 4100, the execution price of the fourth class is 4100 and 4150, and

Table 2. The way features are combined and clustered results

The Model of Encoding	The way features are combined	Num of Clustering
Contrastive-Learning(a)	<years of time, bid-ask spread, bid-ask price ratio>	2
CL-TCN(b)	<years of time, bid-ask spread, bid-ask price ratio>	4
CL(c)	<moneyness, bid-ask spread, bid-ask price ratio>	4
CL-TCN(d)	<moneyness, bid-ask spread, bid-ask price ratio>	5

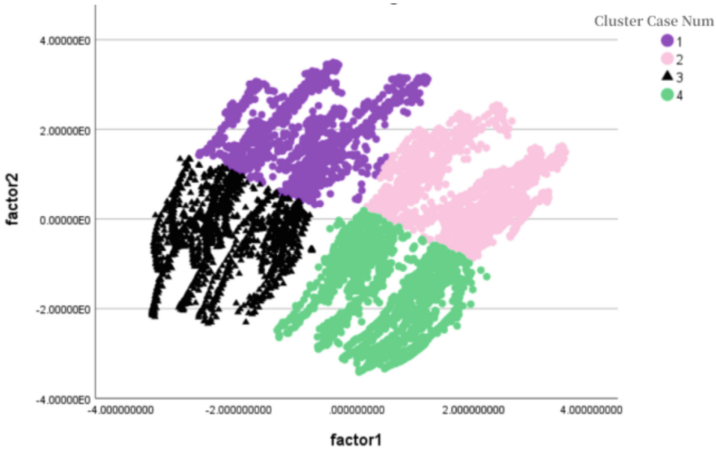


Fig. 9. Plot of clustering$^{(a)}$.

the execution price of the fifth class is 4200. And from the effect point of view, the differentiation between features is stronger.

4.3 Analysis Conclusions of Experimental Results

From this, we find that LSTM or Attention-based models may not achieve the desired effect if they rely on LSTM or Attention-based models alone, but LSTM and Attention-based can play a better role if they rely on the coding generated by CL-TCN. At the same time, the expiration date and strike price, which are closely related to the implied volatility of options, can reflect their impact on the coding effect with the help of CL-TCN, thereby helping traditional time series models to give better prediction conclusions. However, in terms of overall prediction time, the combination effect of CL-TCN and LR is not obvious, so it cannot be completely separated from the LSTM model alone.

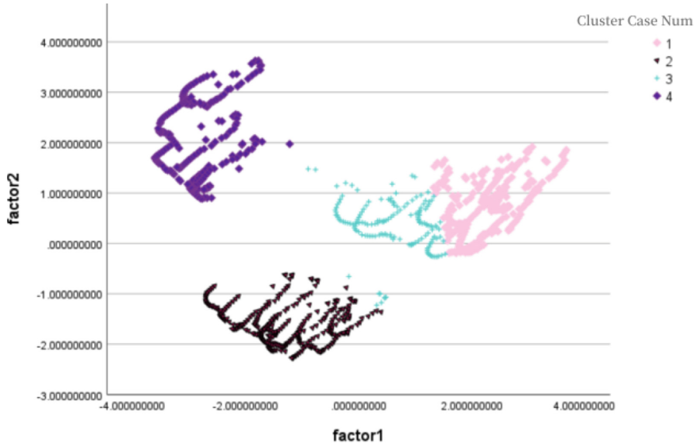


Fig. 10. Plot of clustering(b).

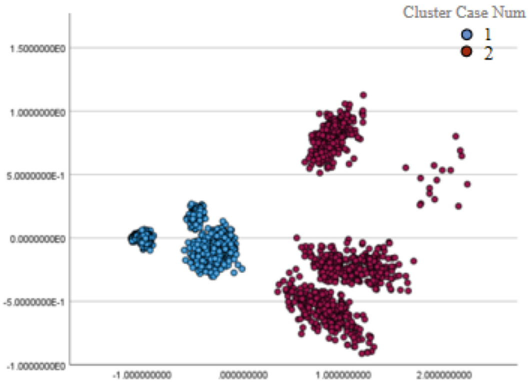


Fig. 11. Plot of clustering(c).

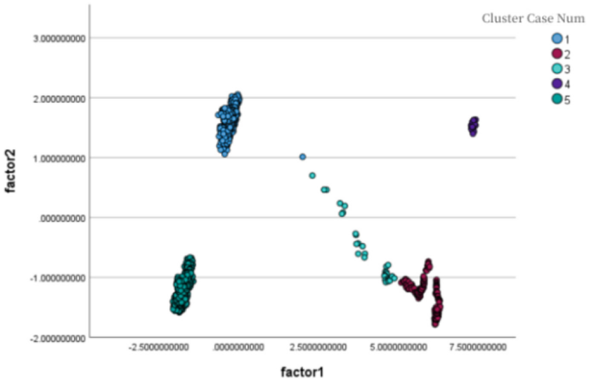


Fig. 12. Plot of clustering(d).

5 Conclusions

This paper practices a method of contrastive learning combined with TCN model for time series vector coding to process fine-grained, high-frequency option market data with obvious business characteristics, and its special timing coding method is of great help to the downstream implied volatility prediction task. We also excavates the business features that are strongly related to options, and uses the clustering algorithm to verify the endogeneity of the model. The disadvantage of this paper is that it cannot be separated from the traditional time series model, it does not solve the problem of high-frequency prediction efficiency, and there is still a huge space for exploration.

References

1. Shi X, Chen Z, Wang H, et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting[J]. Advances in neural information processing systems, 2015, 28.
2. cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.
3. Wang Q, Hao Y. ALSTM: An attention-based long short-term memory framework for knowledge base reasoning[J]. Neurocomputing, 2020, 399: 342-351.
4. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
5. Hewage P, Behera A, Trovati M, et al. Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station[J]. Soft Computing, 2020, 24(21): 16453-16482.
6. Che Z, Purushotham S, Cho K, et al. Recurrent neural networks for multivariate time series with missing values[J]. Scientific reports, 2018, 8(1): 1-12.
7. Chapi K, Singh V P, Shirzadi A, et al. A novel hybrid artificial intelligence approach for flood susceptibility assessment[J]. Environmental modelling & software, 2017, 95: 229-245.
8. Neil D, Pfeiffer M, Liu S C. Phased lstm: Accelerating recurrent network training for long or event-based sequences[J]. Advances in neural information processing systems, 2016, 29.
9. Song H, Rajan D, Thiagarajan J, et al. Attend and diagnose: Clinical time series analysis using attention models[C]//Proceedings of the AAAI conference on artificial intelligence. 2018, 32(1).
10. Tan Q, Ye M, Yang B, et al. Data-gru: Dual-attention time-aware gated recurrent unit for irregular multivariate time series[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(01): 930–937.
11. Eldele E, Ragab M, Chen Z, et al. Time-series representation learning via temporal and contextual contrasting[J]. arXiv preprint arXiv:2106.14112, 2021.
12. Franceschi J Y, Dieuleveut A, Jaggi M. Unsupervised scalable representation learning for multivariate time series[J]. Advances in neural information processing systems, 2019, 32.
13. Yue Z, Wang Y, Duan J , et al. TS2Vec: Towards Universal Representation of Time Series [J]. 2021.