



A multi-feature stock price prediction model based on multi-feature calculation, LASSO feature selection, and Ca-LSTM network

Xiao Chen, Lei Cao, Zhi Cao & HongWei Zhang

To cite this article: Xiao Chen, Lei Cao, Zhi Cao & HongWei Zhang (2024) A multi-feature stock price prediction model based on multi-feature calculation, LASSO feature selection, and Ca-LSTM network, Connection Science, 36:1, 2286188, DOI: [10.1080/09540091.2023.2286188](https://doi.org/10.1080/09540091.2023.2286188)

To link to this article: <https://doi.org/10.1080/09540091.2023.2286188>



© 2024 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 19 Jan 2024.



Submit your article to this journal [↗](#)



Article views: 2987



View related articles [↗](#)



View Crossmark data [↗](#)



Citing articles: 4 View citing articles [↗](#)



A multi-feature stock price prediction model based on multi-feature calculation, LASSO feature selection, and Ca-LSTM network

Xiao Chen, Lei Cao, Zhi Cao and HongWei Zhang

College of Information Engineering, Shanghai Maritime University, Shanghai, People's Republic of China

ABSTRACT

This paper addresses the crucial realm of stock price prediction, highly coveted by individual investors and institutions for its substantial economic implications. The inherent non-stationary and intricate nature of stock market fluctuations, coupled with real-time transactions, poses a formidable challenge for accurate and swift prediction. Unlike prevailing research that predominantly focuses on forecasting methods, our novel approach places a paramount emphasis on processing original data, introducing 57 technical indicators to better represent economic aspects for stock price prediction. Signifying the importance of each feature, we employ the LASSO algorithm to derive an optimal feature combination. Additionally, our methodology utilizes the Ca-LSTM (cascade long short-term memory) technique, enhancing information extraction from individual features. Experimental results, gauged by mean error, underscore the superiority of the Ca-LSTM model over other time series prediction models and conventional long short-term memory approaches. Notably, our model's integration with the accumulation-based VMD-LSTM model demonstrates enhanced forecasting accuracy. This proposed method holds considerable potential to refine stock price prediction, thereby delivering heightened value to investors in the dynamic financial landscape.

ARTICLE HISTORY

Received 4 December 2022

Accepted 16 November 2023

KEYWORDS

Stock; long short-term memory; feature selection; predict; sliding window

1. Introduction

The stock market is a crucial component of the investment market, attracting significant attention from investors and enterprises alike. In the financial field, stock price prediction is an essential and challenging task. It is believed that researches on stock price prediction are of great deal of theoretical significance and application potential (Ding & Qin, 2020). Stock prices exhibit nonlinear, volatile (Bontempi et al., 2012), high noise levels and data intensity (Chang et al., 2009), and posing significant obstacles to accurate prediction. For investors, accurate predictions could bring benefits. Thus people could properly anticipate the company's situation and make informed decisions (Soni et al., 2022). Current studies

CONTACT Lei Cao lciao@shmtu.edu.cn College of Information Engineering, Shanghai Maritime University, 1550 Haigang Avenue, Pudong New Area, Shanghai 201306, People's Republic of China

© 2024 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

have proved that it is reasonable to identify time sequence laws from the past data and make a benchmark forecast of future stock prices. In order to effectively reduce the investment risk and obtain stable investment returns, researchers have explored a number of methods, for example, Random Forest (RF), Support Vector Machine(SVM), Native Bayes, K-Nearest Neighbor (KNN), Softmax, RNN, LSTM , etc.

In the field of financial time series prediction, deep learning methods have gained widespread acceptance due to their superior performance compared to traditional econometric models, and their ability to build forecasting models without expert knowledge (Hoseinzadeh et al., 2022). A study found that deep learning models can extract non-linear relationships between input and output data (Sohani et al., 2021) and it outperformed linear models on the basis of both statistical and economic criteria (Matias & Reboredo, 2012). Based on the literature review of financial time series prediction, Qiu et al. (2020) draw a conclusion that the deep learning models have good nonlinear data fitting ability and adaptive self-learning ability, which is suitable for data with randomness and non-smoothness. This highlights the effectiveness of deep learning methods for forecasting tasks in the financial domain.

Early deep learning methods have proved to have the advantage of fitting nonlinear data. However, early deep learning methods had a limitation of over-fitting data (Wang, Cui et al., 2022). This limitation was overcome with the introduction of recurrent neural networks (RNN), which not only learns the current features of the temporal data through fully connecting hidden neurons but also considers the features gathered by the internal feedback mechanism in the past. But, RNN has disadvantages, including the vanishing gradient problem in long-term data. In order to solve this problem, the long short-term memory(LSTM) network is proposed, which introduces the memory cell to replace the hidden layer memory unit in RNN, as data flows through the network, the output of each layer depends not only on the past state but also on the current weights. Several studies have demonstrated the feasibility of using LSTM networks to predict stock prices (Chen & Ge, 2019; Kumar et al., 2021; Sun et al., 2021). Furthermore, there are two important aspects of long short-term memory are the unit-level neural organisation and the encoding of information (Su et al., 2021). Researchers must design experiments that meet these two requirements. Among them, the parameters of the model are changeable, researchers have also come up with their own methodologies. Parameter adjustment can start from various aspects, that is to say, the number of layers (Gao et al., 2021), the time step and the weight of attention mechanism (Kumar et al., 2021). Although deep neural network can capture data features of complex stock price data, the original features of opening price, closing price, highest price, lowest price and trading volume may not well meet the input requirements of the model. Studies have conducted several comparison experiments, including the comparison of each combination of features, the remaining stocks after filtering the stock according to the price fluctuation (Song et al., 2019), the input features and each combination of various deep learning models. These studies have led to the conclusion that novel input features are necessary for achieving more precise predictions of stock price fluctuations.

The innovations and contributions of this study are as follows:

- (1) The stock technical indicators are calculated based on stock price data, and 25 technical indicators were calculated to reach the final 57 characteristics. Feature combination

screened by LASSO was input into the cascaded LSTM, in order to obtain a short training time and high-accuracy model.

- (2) Proposed a cascaded LSTM model, to accept multiple features data separately. And the model performs superior to the XGboost, Sequential LSTM model. When the cascade model is applied in the VMD-LSTM model, it also performs better.
- (3) To improve the accuracy of the proposed model, the rule of time division is studied. Time series data is divided into four groups, experimental results prove that the accuracy of the model increases with the input of different time interval time series data.

2. Related works

In recent years, deep learning has flourished, and it has been shown to be effective in dealing with time series data, including financial data. While there is a focus on developing effective deep learning models for financial analysis, it is important to recognise that high-quality data is equally critical for achieving accurate results. Prior to entering the network, data must be pre-processed, to ensure that it is in a suitable form for analysis. Therefore, careful consideration of data quality and appropriate pre-processing techniques are essential for the success of deep learning applications in finance.

Forecasting models often benefit from multiple features including transactional data, news, social media and search behaviour (Li et al., 2020). According to Wang, Cui et al. (2022), the analysis of multiple factors has been found to offer more effective information and improve prediction accuracy in stock price prediction. Furthermore, investors are increasingly inclined to utilise multi-source data for decision-making purposes (Wu et al., 2022), primarily due to its interpretability. Moreover, some models cannot extract and recognise features well by themselves and require additional features pre-processing (Li et al., 2023). And methods could be feature selection algorithms or an integrated network structure that can extract features for specific data (Zhao & Yang, 2023). However, including multiple features can also introduce noise and increasing the training time. For example, inappropriate introduction of positive and negative sentiment indicators leads to poor prediction results. How to solve this problem? In the field of time series prediction, an effective method is to select the optimal feature combination, Li et al. (2022) observed the holdings and returns of Hong Kong-funded institutions over the years, found that different institutions predicted and preferred stocks differently, given their diverse trading characteristics. Multi-objective optimisation algorithms are also widely used to find the optimal solution from a large number of features and parameters (Mahmoudan et al., 2022, 2021; Sohani, Pedram et al., 2021). In terms of stock price prediction, different feature screening methods have advantages and disadvantages. And it is believed that classic linear mapping dimensional reduction methods such as principal component analysis (PCA) and discriminant analysis (LDA) cannot get good results for nonlinear problems and were not suitable for dimensional reduction, which affected the multi-dimensional indicators of stock prices. Recently, linearisation of nonlinear stream learning (LML) techniques has led to advances in effective and efficient algorithms (Chen et al., 2022), and the nonlinear dimensional reduction method (Locally Linear Embedding LLE) could break through the limitations of the principal component analysis method (Yu et al., 2020) in nonlinear data. In addition to selecting methods based on the linear and nonlinear nature of the data, researchers

also consider methods that extract and then cluster the data, thus making them more suitable for time series prediction models, such as using the XGboost method to extract features from the original data, then clustering them using the K-means method, and finally inputting the data into a deep neural network model (Wang & Zhu, 2022). Although this approach has some advantages, such as better performance than other methods, but it still has limitations associated with the difficulty in determining screening parameters, incomplete representation of information, and long training time. In terms of evaluation metrics, the error values are also higher compared to other literature.

For the selection of prediction method, methods used by researchers are constantly changing. Traditional financial time series forecasting methods, such as auto regressive moving average model (ARMA), generalised auto regressive conditional heteroskedasticity (GARCH) (Bollerslev, 1986), which require high smoothness of the data and have been supplemented by machine learning techniques, including support vector machine model (SVM) (Xia Y used in stock price predicting; Xia et al., 2013), and artificial neural network (ANN). Moreover, an increasing number of researchers have been exploring the use of deep learning models in combination with data screening methods. In addition to PCA and LLE dimensional reduction methods mentioned above, researchers have combined the VMD (variational mode decomposition) method with deep neural networks. Yujun et al. (2021) used VMD method and EEMD (Ensemble empirical mode decomposition) method to divide the index price into components with different bandwidths named intrinsic mode functions (IMF). They then inputted each IMF into LSTM model, respectively, since the sum of IMFs was the same as the original value, which had certain significance, and writer pointed out that the method was based on the idea of dividing and solving complex problems. Most of the methods based on modal decomposition will use independent architectures and have been proven effective in the results (Fu et al., 2022). Although the VMD method used by the researchers has improved the fit of the model to the time series to some extent and made the input data more suitable (Niu et al., 2020), there are still areas of deficiency, mainly in the following two aspects, (1) the method mainly performs the modal decomposition of a single time series (i.e. a single feature, mostly using the closing price), thus ignoring the influence of other features and causing the loss of information. (2) The components predicted by the model often derive the final results by accumulation, leading to the introduction of noise and imprecision of the results.

For the inner structure of deep learning network, researchers are constantly exploring new ways to optimise the structure of deep learning networks, and one popular approach is to experiment with different parameter combinations or layer structures in the network. While RNN/LSTM network lacks the advantage of stopping early (Kumar et al., 2021), limiting the number of epochs could prevent the model from over-fitting to the training data. and by controlling the number of recursions, the LSTM can be adapted to infinite scenarios with different timing rules (Tang et al., 2022). Additionally, a well-designed network should use proper activation functions and parameters to enhance performance. And when it comes to the structure, attention-based models, especially classification LSTM models, have shown great success (Yu & Kim, 2019). Hollis et al. (2018) combined LSTM network with ATTENTION mechanisms, and pointed out that the idea was to leverage the developments in attention mechanisms to improve the performance of promising LSTM RNN architectures currently in use for FTS (Fuzzy Time Series) forecasting. Experimenting with the loss function as the model hyperparameter adjustment criterion, the ATTENTION LSTM obtains 60

% in prediction, which is higher than the 58 % of the baseline LSTM. It is worth noting that the method used in this paper is sensitive to hyperparameters, and small adjustments can lead to very different results, which can have a huge impact on model training. Another successful example is the work (Li et al., 2022), which put the ATTENTION-based LSTM network into the framework of migration learning and adopted the idea of adversarial learning, and finally obtained a good performance.

Based on the research above, we have focused on the input data and addressed the issue of insufficient features in the previous studies by introducing techno-economic indicators. We have also employed feature selection methods and optimised feature combinations. Moreover, an architecture similar to the VMD approach, namely, Ca-LSTM network, is proposed to introduce other features that VMD fails to introduce.

3. Methodologies

In this paper, the LASSO feature selection method is combined with the cascaded LSTM network to improve the precision of stock price prediction. This section introduces the basic principle of each method employed in the proposed model.

3.1. LASSO

The LASSO method is a regression analysis method that simultaneously selects features, aiming at enhancing the prediction accuracy and interpretability of statistical models. It is a penalised least square method and imposes an L1-penalty on the coefficients (Tibshirani, 1996). It is considered a constrained optimisation issue that intend to make the absolute weights being less than a constant t (Coelho et al., 2020), as what is expressed in the formula. The formula has y as the dependent variable, x as the independent variable, α as the regression model constant, and β as the coefficient of P independent variables. The t on the right-hand side represents the penalty coefficient, when the $t < t_0$, some coefficients in the regression model will become 0 and be eliminated, thus achieving the effect of feature screening.

$$(\hat{\alpha}, \hat{\beta}) = \arg \min \left\{ \sum_{i=1}^N \left[y_i - \alpha - \sum_{j=1}^p \beta_j x_{ij} \right]^2 \right\}, \quad S \cdot T \cdot \sum_{i=1}^P |\beta_i| \leq t \quad (1)$$

3.2. LSTM

As a time series data-friendly model, LSTM has been widely applied in the forecasting fields, as a variant of RNN, it solves the problem of gradient explosion. Additionally, its unique architecture (Figure 1) allows it to better retain or forget temporal data information than other deep learning models in terms of retaining or forgetting temporal data information.

As Figure 1 shows, LSTM model contains three gates: input gate (i_t), forgetting gate (f_t) and output gate (o_t). The input gate is used to determine whether a new input should be added to the layer, the forgetting gate is used to determine whether past information is discarded from the layer and the output gate is mainly used for output data. The main

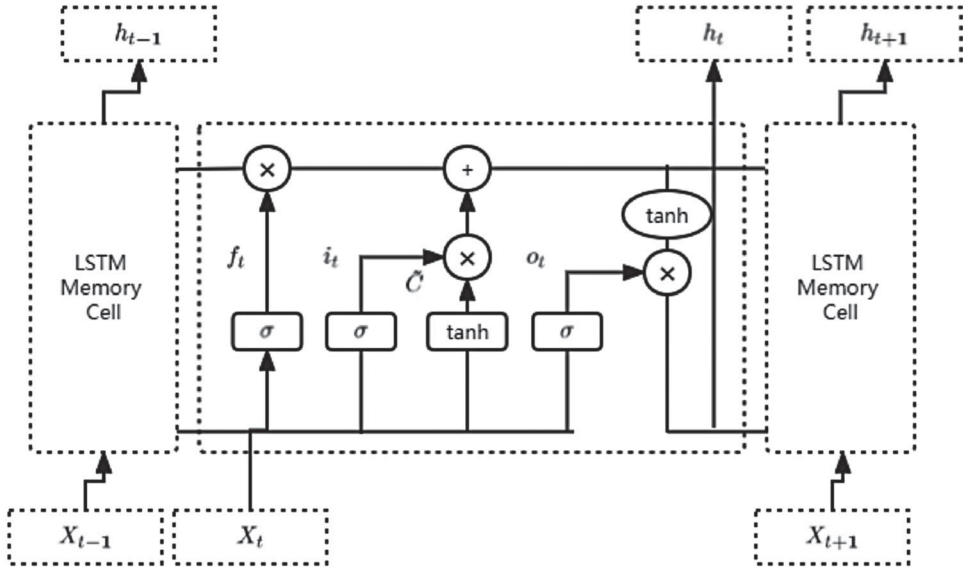


Figure 1. Structure of LSTM network.

processes of the LSTM model can be given as follows:

$$i_t = \sigma(\omega_i, k[h_{t-1}, x_t] + b_i) \quad (2)$$

$$f_t = \sigma(\omega_f * [h_{t-1}, x_t] + b_f) \quad (3)$$

$$o_t = \sigma(\omega_o * [h_{t-1}, x_t] + b_o) \quad (4)$$

$$C_t = f_t * C_{t-1} + i_t * C_t \quad (5)$$

$$C_t = \tanh(W_c * [h_{t-1}, x_t] + b_c) \quad (6)$$

$$h_t = o_t^* \tanh(C_t) \quad (7)$$

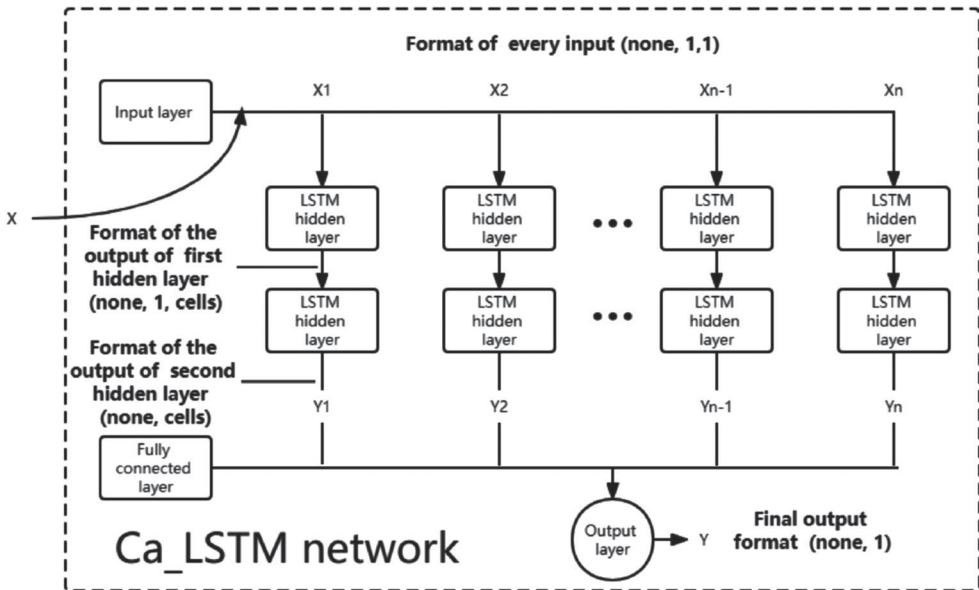


Figure 2. Structure of Ca-LSTM.

3.3. Ca-LSTM

Based on LSTM, we propose a cascade LSTM model (Figure 2). In this model, the input gate is divided into six gates for receiving independent features. After data is fully calculated by the two LSTM hidden layers, we add a fully connected layer to store the results and finally pass the results to the output layer. We chose to use 80 neurons in each hidden layer, as this number performed well on most datasets when we manually adjusted the parameters. And

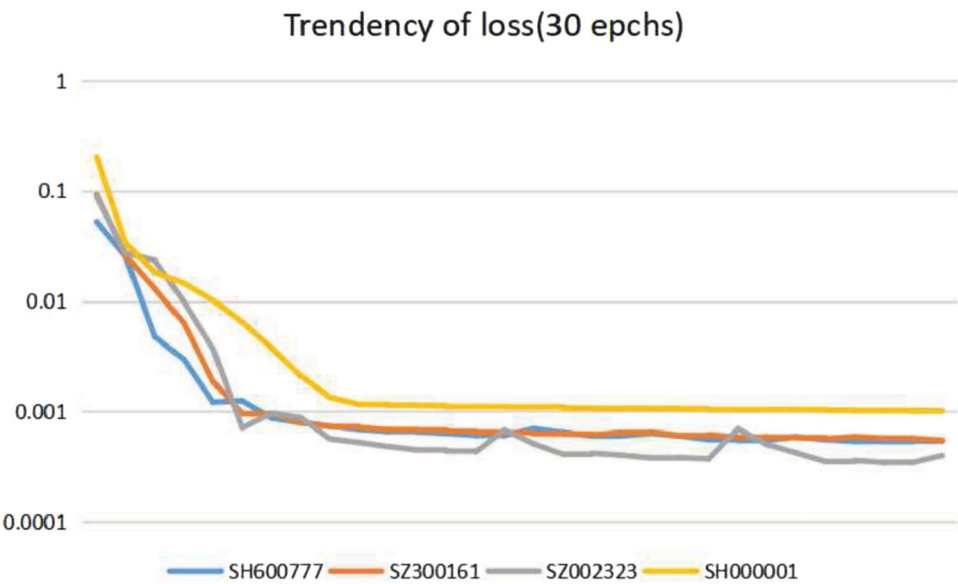


Figure 3. Tendency of loss (30 epochs).

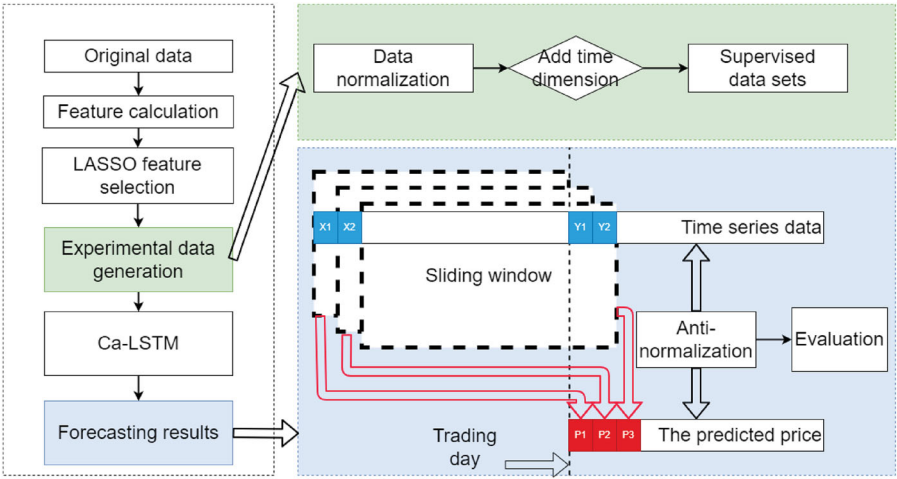


Figure 4. Flow chart of the LASSO + Ca-LSTM model.

after the full connection layer, we used the Adam optimiser, which is a variant of stochastic gradient descent, to optimise the model's parameters during the training.

The training process of a stock prediction model varies in its fit across different stocks. Applying the same parameters to different stocks can lead to overfitting or underfitting, especially when forecasting the SH000001 (Shanghai Composite Index) with its high stock price. To ensure accurate forecasts, we use loss tendency as the evaluation criterion and adjust parameters when the loss no longer shows a downward trend. In our study, we employed manual hyperparameter tuning based on loss tendency and performance on test datasets for various stocks. This optimisation approach allowed us to address overfitting or underfitting issues and improve model performance. We adjusted the number of epochs, batch size, activation function, and dropout rate based on the model's validation set performance. During the training, we closely monitored loss and accuracy metrics, stopping when the model reached a performance plateau on the validation set, preventing overfitting and ensuring generalisation. To comprehensively assess model performance, we compared predictions on validation datasets across different stocks using evaluation metrics such as MSE, RMSE, MSLE, and MAPE. This multi-metric evaluation provided a comprehensive understanding of the model's predictive capabilities. By employing manual hyperparameter tuning and comprehensive evaluation, we aimed to strike a balance between model complexity and generalisation, ultimately enhancing the accuracy and robustness of our stock prediction model, particularly for challenging stocks like the SH000001 (Figure 3).

After the model is fully trained, the sliding window idea in the prediction link can better simulate the real trading scenario by inputting the data of the last three to four months to predict the stock price at the next point in time, and using the point in time as the sliding range, waiting for the end of the point in time to continue to add the stock price at the next point in time to correct the model, and so on and so forth, the specific process we show in Figure 4.

3.4. Description and transformation of data

The original data contains stocks issued from the Shanghai stock exchange and Shenzhen stock exchange. Each stock has five features, including four stock price features and one quantitative feature, and the time period is from 2016 to 2020. In order to make a better selection, I choose volatility (as follows, p refers to price) as the stock selection criterion, volatility could better reflect sequence characteristics. There is a significant relation between volatility and expected stock returns (Bali & Hovakimian, 2009). Finally, I selected 30 stocks as experimental data set as is shown in Table 1. In order to better explore the data, technical indicators are calculated, which are used for stock evaluation in economics based on their original five characteristics, a total of 52 items, as is shown in Table 2, it is worth noting that since some of the items in the table contain sub-items, the final number of features is 52. We then take the original five features into account, and the final number of features in the constructed dataset is 57.

$$R = r_t = \frac{p_t - p_{t-1}}{p_{t-1}} \quad (8)$$

Table 1. The datasets of the 30 selected stocks from 1 January 2016 to 30 November 2020.

Lowest		Middle		Highest	
Stock code	Volatility	Stock code	Volatility	Stock code	Volatility
SH600777	21.29633380	SH600277	51.91269969	SZ000550	132.0974239
SH603069	21.45187148	SZ002122	52.35519146	SZ002559	134.7197498
SH600328	22.65305280	SH600618	52.51904444	SZ300488	138.4139677
SH601168	22.70975596	SH601678	52.58151945	SZ002391	160.9385138
SZ002132	23.07880543	SZ300161	52.90560184	SH600146	170.2241102
SH601117	23.16004278	SH600804	53.02546691	SH601777	191.7850969
SH600367	24.62127041	SZ002012	53.74672573	SZ300343	232.8665563
SH600389	24.86752082	SH603338	54.24341316	SH603996	237.7075488
SH601390	24.91422983	SZ002497	54.31482465	SZ002599	260.3293523
SH601127	24.95312135	SH603599	54.35540975	SZ002323	286.8527704

Table 2. The technical indicators of the stocks.

Technical indicator	Description	Technical indicator	Description
MA	Moving average	CCI	Commodity Channel Index
MACD	Moving average convergence divergence	ATR	Average True Range
KDJ	Random indicator	BBI	Long-short indicators
WR	William indicators	DMI	Trend indicators
BTAS	Good rate	TAQ	Down Anchi Passage
BOLL	Bollinger band	TRIX	Triple exponentially smoothed average
PSY	Psychological Line	ROC	Index of rate of change
OBV	Energy tide index	ASI	Vibration lifting index
MFI	Money Flow Index	XSLI	Shears passage II
VR	Volume ratio	DFMA	Parallel line difference index
EMV	Simple volatility indicator	MTM	Momentum index
DPO	Interval oscillation line	MASS	Mace line
EXPMA	EMA Index averages		

$$Volatility = \sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (r_i - \bar{r})^2} \quad (9)$$

After fully calculating the data, a set of characteristics describing the stock can be obtained, but for the LSTM network, the selection of features affects the accuracy, the training time and the value of the output data of the model. Feature differences are also very large. And feature selection could prevent the selection of too many or too little features (that is, more or less than necessary), and implement data reduction to speed up training and improve computational efficiency (Piramuthu, 2004). The features that are negatively correlated with the regression label features will make the effect of the model become extremely poor, while the features of low correlation will lead to longer training time, even if they improve the model's accuracy. This is a significant disadvantage in today's world of real-time trading. Therefore, the selection of features is extremely important. In this study, LASSO method is used to screen features with weights greater than 0, with the aim of obtaining the optimal combination of features as an input of LSTM network training (Table 3).

$$Up = \max (x_{t-n}, x_{t-n+1} \dots x_t) \quad (10)$$

$$Low = \min (x_{t-n}, x_{t-n+1} \dots x_t) \quad (11)$$

$$mid = avg(Up, Low) \quad (12)$$

$$VR = \left(\sum_1^N (IF (price_t - price_{t-1} > 0)), 1, 0 \right) \\ + 1/2 * \sum_1^N (IF (price_t - price_{t-1} = 0), 1, 0) \quad (13)$$

$$TYP = avg(HIGH, LOW, CLOSE) \quad (14)$$

$$V1 = SUM(IF(TYP \geq REF(TYP, 1), TYP * VOL, 0), N) \\ / SUM(IF(TYP \leq REF(TYP, 1) TYP * VOL, 0), N) \quad (15)$$

$$MFI = 100 - (100 / (1 + V1)) \quad (16)$$

To reduce the impact on noise and facilitate optimisation of the solving process, each sample of data set is normalised to the range of [0, 1] by the following maximum and minimum standardised method.

$$x(t)' = (x(t) - \min x(t)) / (\max x(t) - \min x(t)) \quad (17)$$

To apply the LSTM network as a regression model, it is necessary to transform the original time series data including samples and features in supervised learning and add a time dimension, so that the predicted value can be obtained after it is input into the LSTM network. We add one timestep to the original data, and transform it as follows:

$$\begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_t \end{bmatrix} \Rightarrow \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_{t-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_t \end{bmatrix} \quad (18)$$

$$\begin{bmatrix} x_{00} & x_{01} & \dots & x_{0n} \\ x_{10} & x_{11} & \dots & x_{1n} \\ \dots & \dots & \dots & \dots \\ x_{t0} & x_{t1} & \dots & x_{tn} \end{bmatrix} \Rightarrow \begin{bmatrix} x_{00} & x_{01} & \dots & x_{0n} \\ x_{10} & x_{11} & \dots & x_{1n} \\ \dots & \dots & \dots & \dots \\ x_{(t-1)0} & x_{(t-1)1} & \dots & x_{(t-1)n} \end{bmatrix} \begin{bmatrix} x_{10} & x_{11} & \dots & x_{1n} \\ x_{20} & x_{21} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{t0} & x_{t1} & \dots & x_{tn} \end{bmatrix} \quad (19)$$

As the model takes in normalised time series data, the output (i.e. predicted value) of the model is also normalised. In order to obtain the actual predicted stock prices, it is necessary to perform inverse normalisation on the output. This can be done by multiplying the predicted value by the range of the original stock price and adding the minimum stock price. This will give us the actual predicted stock price, which is on the same scale as the original data.

$$x(t) = x(t') (\max x(t) - \min x(t) + \min x(t)) \quad (20)$$

Table 3. Features remaining after using the LASSO method.

Feature name	Description of features	Value of LASSO-based weight
Angel tang passage (TAQ)	Formula (10)(11)(12)	0.00703028
Volume ratio (VR)	Formula (13)	0.00025557
Money Flow Index (MFI)	Formula (14)(15)(16)	0.00017774
High	Original features	0.2330305
Low	Original features	0.06854353
Close	Original features	0.67546297

3.5. Evaluation criteria of experimental results

A good way to measure the effect of a model's time series output is to mean square it with the actual stock price (i.e. the test set). However, for the output results of the model, it is not enough to do the fitting, for stock price forecasting, the next time the price of joining can make the changes as a result, so we adopt the way of sliding window in the model to predict the shares of a point in time, after sliding data, to join the next time point in the model's price, time share prices are obtained. Here we predict the stock price at 30 points and then compare the predicted price with the actual stock price as an evaluation metric. Specifically, we calculate several commonly used metrics, RMSE (root mean square error), MAE (mean absolute error), MAPE (mean absolute percentage error), MSLE (mean squared logarithmic error) as follows:

$$RMSE = \sqrt{(1/N) * \sum_{t=1}^N (x_t - \hat{x}_t)^2} \quad (21)$$

$$MAE = \frac{1}{N} * \sum_{t=1}^N |x_t - \hat{x}_t| \quad (22)$$

$$MAPE = \frac{1}{N} * \sum_{t=1}^N \left| \frac{(x_t - \hat{x}_t)}{x_t} \right| \quad (23)$$

$$MSLE = (1/N) * \sum_{t=1}^N (\log(\hat{x}_t + 1) - \log(x_t + 1))^2 \quad (24)$$

3.6. Model complexity and overhead analysis

In this section, we provide a comprehensive analysis of the complexity and overhead associated with our proposed Ca-LSTM network for stock price prediction. We discuss the computational requirements at different stages of our approach, including data pre-processing, model training, and prediction. Additionally, we examine the practical implications of implementing the Ca-LSTM model, such as computational time and memory consumption.

To evaluate the computational complexity of our proposed approach, we analyse the time and space complexity at each stage of the Ca-LSTM model. Table 4 presents a summary of the complexity analysis.

In the data preprocessing stage, tasks such as data normalisation, feature engineering, and sequence partitioning have a linear time complexity of $O(N)$, where N represents the

Table 4. Complexity analysis of the Ca-LSTM model.

Stage	Time complexity	Space complexity
Data preprocessing	$O(N)$	$O(1)$
Model training	$O(E * S * L)$	$O(L * H)$
Prediction time	20 ms	
Training time	139 ms/step	

number of data points in the input sequence. The space complexity remains $O(1)$, as no additional storage requirements are introduced in this step.

For model training, the Ca-LSTM network involves forward and backward propagation, weight updates, and gradient computations. The time complexity is proportional to the number of epochs (E), the number of samples (S), and the number of layers (L). Hence, the overall time complexity is approximately $O(E * S * L)$. The space complexity depends on the number of layers and hidden units, resulting in a space complexity of $O(L * H)$, where H represents the number of hidden units in each layer.

During the prediction phase, the Ca-LSTM model performs forward propagation to generate future price predictions. The time complexity for prediction is $O(S * L)$, where S denotes the number of steps in the input sequence. The space complexity remains $O(L * H)$, similar to the training phase.

To assess the prediction time overhead of the Ca-LSTM model, we measured the average time required to generate predictions compared to the LSTM method. Due to the nature of the cascade, the training time of the Ca-LSTM method is roughly a constant multiple of that of the LSTM method.

In summary, our analysis demonstrates that the Ca-LSTM model exhibits a reasonable computational complexity throughout the various stages of data preprocessing, model training, and prediction.

4. Experiment description and results

4.1. Comparison of major forecasting models

In this part, two types of experiments are designed to evaluate the performance of the cascade LSTM network. Firstly, the LSTM network is compared with other time series data to predict the network effects, in order to get the cascade way LSTM network superiority. Secondly, the input data is divided into three levels: 1 h and 30 min, 10 min, to test the cascade LSTM network the circumstances under the optimal performance.

Table 5 shows the experimental results predicted by the model of 3 stocks, 1 index and the average value of 30 stocks. We compared our proposed model with various existing models, and all results were obtained from the same dataset used for our proposed model. For each model compared, we referred to the literature to replicate the model's structure and parameter adjustment methods to ensure a fair comparison of the results. Table 4 includes a range of commonly used prediction methods, spanning various fields of study. Specifically, ARMAX is a popular statistical approach, while SVR and XGBOOST are widely used in the field of machine learning, respectively. The deep learning methods of BP and LSTM are also included. Besides, since Ca-LSTM is also based on the same gate approach to overcome the shortcomings of RNN, in order to compare deep learning models from the gradient perspective, we refer to a gradient-free model PSO-SRNN based on particle swarm

Table 5. Predictive performance evaluation of different models of different stocks.

Studies	2016	2018	2022	2013	2011	1997	2022	Our research	2015	2020
Methodologies	LASSO								PCA	LLE
Indicators	ARMAX	XGBOOST	SVR	BP	LSTM	PSO-SRNN	Ca-LSTM		Ca-LSTM	Ca-LSTM
SH600777										
RMSE	0.07	0.46	0.15	0.43	0.13	0.22	0.06		22.52	0.46
MAE	0.05	0.46	0.14	0.42	0.127	0.21	0.05		17.07	0.45
MAPE	2.32	28.1	6.68	16.75	5.74	11.60	2.36		302.73	18.05
MLSE	0.0004	0.03	0.002	0.02	0.002	0.006	0.004		1.26	0.02
SZ300161										
RMSE	0.73	11.46	0.99	0.95	0.62	0.51	0.58		28.83	2.59
MAE	0.55	11.44	0.88	0.79	0.51	0.43	0.8		24.2	2.38
MAPE	3.56	264.83	5.92	5.19	3.34	2.78	3.13		343.61	18.82
MLSE	0.002	1.32	0.004	0.003	0.001	0.001	0.001		3.61	0.03
SZ002323										
RMSE	0.53	1.38	1.49	0.45	0.47	0.44	0.41		56.98	1.85
MAE	0.44	1.28	1.35	0.39	0.4	0.37	0.35		42.09	1.8
MAPE	15.26	69.3	29.28	12.91	11.78	13.57	10.44		324.52	49.85
MLSE	0.02	0.14	0.09	0.01	0.01	0.012	0.009		1.29	0.27
SH000001										
RMSE	88.98	2876.47	231.55	94.39	56.89	56.70	56.01		115.54	1111.72
MAE	76.44	2875.53	220.29	84.99	46.07	42.77	44.93		94.51	856.04
MAPE	2.61	4725.72	7.1	2.88	1.6	1.48	1.56		341.82	68.97
MLSE	0.001	14.9	0.006	0.001	0.0004	0.0003	0.0004		4.85	0.36
Average value										
RMSE	0.70	1.94	2.06	2.98	0.52	0.472	0.466		37.63	2.71
MAE	0.62	1.89	1.88	2.80	0.42	0.36	0.36		30.51	2.34
MAPE	5.61	12.57	22.60	15.76	4.78	4.41	3.68		364.82	32.16
MLSE	0.004	0.03	0.07	0.007	0.003	0.003	0.002		3.85	0.42

optimisation (PSO) algorithm in the literature (Bas et al., 2022) and compare it with other models. And to compare feature selection algorithms, we selected LLE, PCA, and LASSO. Notably, LASSO reduced the number of features to six, so we also included ARMAX (a variant of ARMA) with six features for comparison.

We divide our dataset into three subsets: the training set, validation set, and test set. The training set comprises three years of historical stock price data, allowing the model to learn long-term patterns and trends. We allocate six months of data for the validation set, which helps fine-tune the model’s hyperparameters during the training. The remaining six months of data form the test set, serving as an independent benchmark to evaluate the model’s performance on unseen data. Additionally, we utilise a sliding window scheme to create sequential input-output pairs from the time series data, enabling the model to capture both short-term and long-term dependencies, which could also help the model to better simulate a real scene and to eliminate the model’s failure to take new data into account. Take the trading day prediction scenario as an example, the model runs daily to predict the stock price for the following trading day. At the end of the trading day, if the price data generated by the trading is not taken into account, the model will be stuck in the past data, resulting in inaccurate predictions. By adding the data of each trading day through the sliding window, the model can timely adjust its prediction to achieve better accuracy.

The cascaded LSTM (Ca-LSTM) network consistently demonstrates superior performance compared to other methods across all four datasets, showcasing its advantages. Notably,

as the volatility decreases, the Ca-LSTM model's performance further improves. Even in the case of the SH000001 dataset, which exhibits high value and challenging prediction characteristics, the Ca-LSTM model outperforms alternative methods in terms of RMSE and MAE metrics. While exploring the non-gradient Particle Swarm Optimisation-based SRNN (PSO-SRNN) algorithm, we observed its commendable performance surpassing that of the gradient-based LSTM method. Specifically, it excelled in the prediction of the SZ300161 stock set. However, upon evaluating the average performance across the 30 stocks, the PSO-SRNN method slightly underperformed when compared to our proposed Ca-LSTM model. It is worth noting that the PSO-SRNN method carries the drawback of high time overhead. Despite this limitation, it highlights the potential of optimisation algorithms for enhancing model performance. In conclusion, the comprehensive evaluation of our proposed Ca-LSTM model, alongside comparative analyses of other methods, underscores its superior performance across diverse datasets and the average evaluation metrics. While the PSO-SRNN method demonstrates competitive results on one dataset, it falls short of our model's overall performance. Furthermore, the PSO-SRNN method's high time overhead serves as a valuable reminder of the optimisation algorithm's potential for enhancing model outcomes. These findings reinforce the efficacy and potential of the Ca-LSTM model in the field of deep learning stock price prediction.

4.2. prediction performance of Ca-LSTM

And we could also get the predicted curve by the method of sliding window and the actual curve of the three stocks and one SH index as is shown in Figure 5.

The dataset includes three stocks with varying levels of volatility: the lowest, middle, and highest. Upon examining the predictions, it is evident that the first two stocks have more accurate predictions, with predicted prices aligning closely with actual price movements, particularly during the first 12 points in the top left subfigure of Figure 5 and the first 24 points in the top right subfigure of Figure 5. Notably, the model quickly adjusts to the trend after the discrete points in the first two charts, accurately repeating predictions. Conversely, the less stable third chart captures the trend in the actual data only a few days later. It is important to note that while predicting future prices for all four examples, the magnitudes of their prices on the y-axis differ significantly. The first three examples, with smaller price magnitudes, allow for better observation of the model's performance, as the trend of *pre_price* and *true_price* is the same, leading to highly accurate predictions. The fourth example has larger price magnitudes, resulting in larger differences between *pre_price* and *true_price*. Despite this, the overall trend remains consistent, demonstrating the model's value.

In order to assess the performance of the Ca-LSTM model in predicting stock prices over different time horizons, we conducted an experiment by adjusting the length of the prediction interval in the sliding window method. By varying the forecast horizon, we aimed to evaluate the model's capability to generate accurate predictions for short-, medium-, and long-term time series. The experiment involved training the Ca-LSTM model on historical stock price data and evaluating its performance for different prediction intervals. The prediction intervals considered in this experiment were categorised as short-term, medium-term, and long-term, representing 40, 60, and 80 days, respectively.

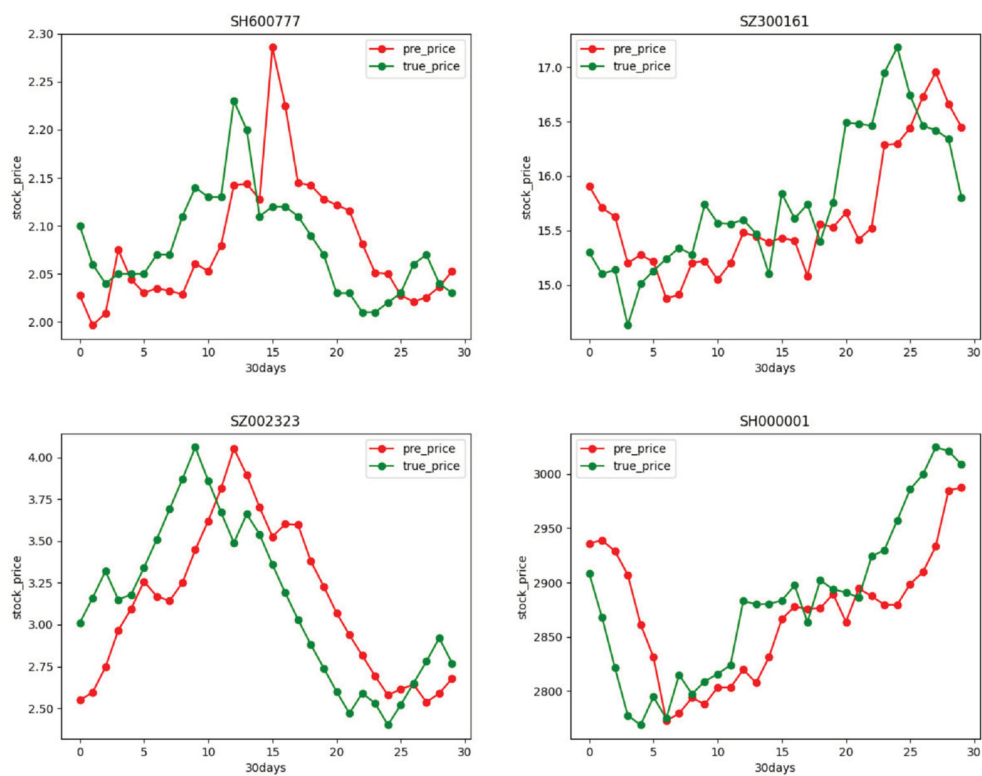


Figure 5. Predicted price of LSTM and sliding window.

Table 6. Predictive performance evaluation of different stocks with different time lengths.

Stock code	Evaluation index	40 days	60 days	80 days
SH600777	RMSE	0.08	0.10	0.19
	MAE	0.05	0.08	0.18
	MAPE	2.51	3.78	9.00
	MSLE	0.0004	0.001	0.0035
SZ300161	RMSE	0.59	0.61	0.64
	MAE	0.47	0.51	0.48
	MAPE	2.75	2.73	2.48
	MSLE	0.0009	0.001	0.0017
SZ002323	RMSE	0.29	0.29	0.36
	MAE	0.23	0.22	0.30
	MAPE	7.36	7.08	9.09
	MSLE	0.005	0.005	0.007
SH000001	RMSE	44.45	47.80	51.26
	MAE	35.12	38.56	40.86
	MAPE	1.20	1.32	1.41
	MSLE	0.0002	0.0003	0.0003

As is shown in Table 6, the experimental results demonstrate that the Ca-LSTM model exhibits consistent and reliable performance across short-, medium-, and long-term time series forecasting. Despite the increased forecast horizon, the model shows only a minimal increase in the error across the four evaluation indicators. These findings also indicate that our model maintains its stability and effectiveness in long-term time series forecasting,

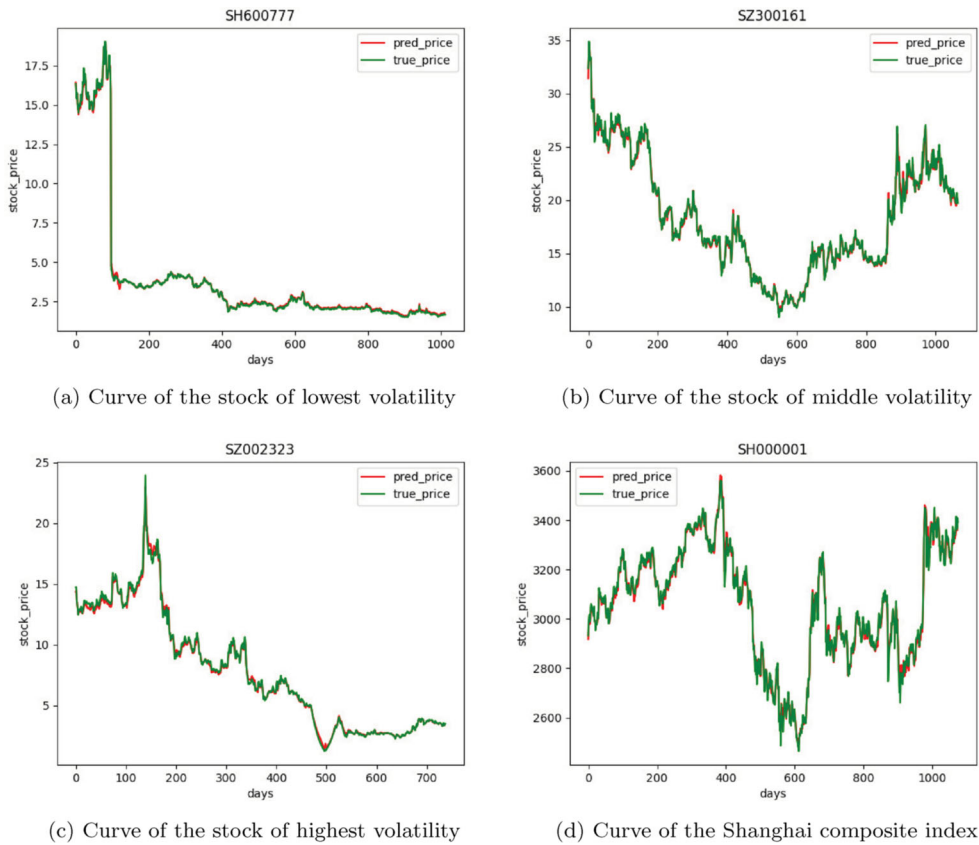


Figure 6. Fit curves of three stocks and one index. (a) Curve of the stock of lowest volatility, (b) curve of the stock of middle volatility, (c) curve of the stock of highest volatility and (d) curve of the Shanghai composite index.

showcasing its ability to capture and forecast price movements over extended periods. The relatively small increase in error suggests that the model successfully captures the underlying trends and patterns in the stock market, enabling it to provide reliable predictions even for longer time horizons.

Fitting curves are useful for visualising the quality of data fitting of the network and understanding the overall trend in data, which is different from the sliding window method mentioned above. Figure 6 shows the prediction results for time series data of the three stocks, it can be seen that the model fits the data well.

The results of the 30 stocks are compared as is shown in Figure 7.

We can roughly divide stocks into three groups, the volatility of these three groups has a big difference outside the group, and a small difference within the group. It can be clearly seen from the figure that, apart from the discrete value, evaluation index has an upward trend with the volatility. For stocks with similar volatility, the evaluation index is also similar.

4.3. Comparison with VMD-LSTM method

The VMD-LSTM (Figure 8), which was introduced in the paper (Niu et al., 2020), presented an idea of decomposition of signal of stock data, enabling the model to learn more effectively.

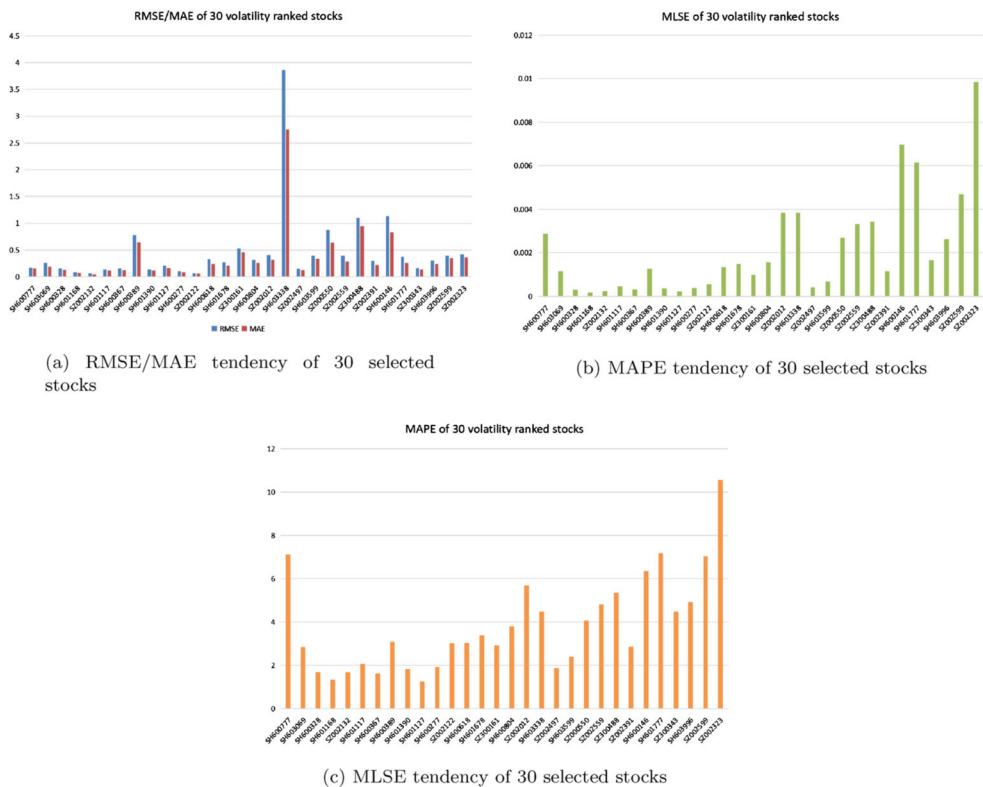


Figure 7. Example of a two-part figure with individual sub-captions showing that captions are flush left and justified if greater than one line of text. (a) RMSE/MAE tendency of 30 selected stocks, (b) MAPE tendency of 30 selected stocks and (c) MLSE tendency of 30 selected stocks.

In this model, each IMF can be predicted by LSTM model, and finally the predicted prices are accumulated to get the stock price.

The method mentioned in this paper is similar to my proposed cascade LSTM structure, so I consider replacing the method in this paper. Table 7 shows the comparison of results before and after replacement. It can be seen that all the metrics of the replaced model are stronger than the original VMD model. The reason why the values of MAPE indicators in Table 7 are all 100 is that, in order to compare with the original paper, the data used for model training and prediction are normalised data as in the original paper, and it is characteristic that all the values are less than 1. Therefore, the values of MAPE indicators, which represent the percentage differences between the predictions and actual values, are all 100. In addition, Table 7 also shows the comparison of effects between the proposed model and other temporal prediction models .

4.4. Performance of the model at different time levels

To better leverage the model’s advantages, I conducted a time-scale experiment and obtained some insights. When we apply the model in the trading scenario, we can consider its advantages of division of time level more. Table 8 shows the performance of the

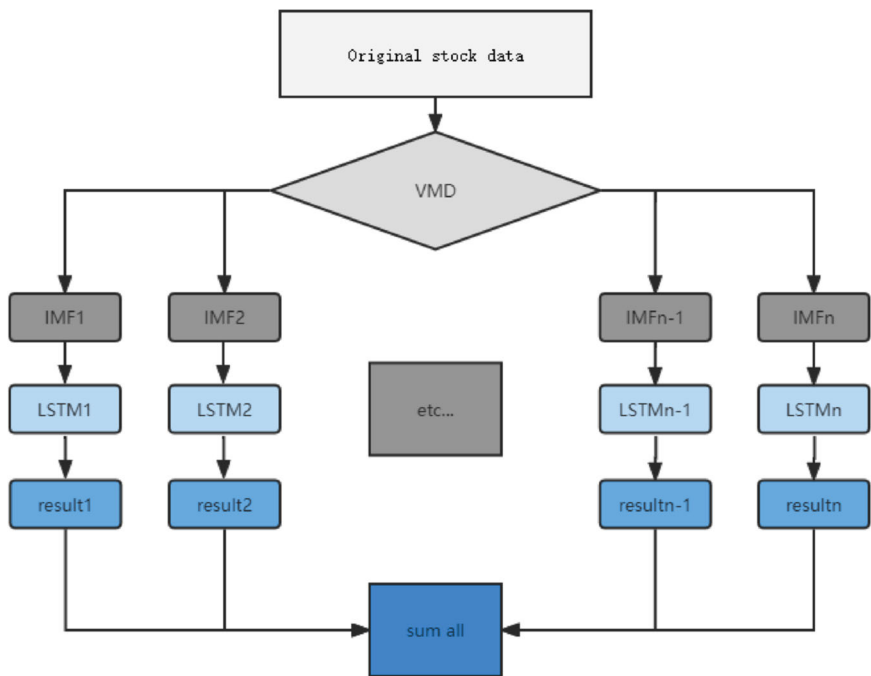


Figure 8. VMD-LSTM Structure of the paper (Niu et al., 2020).

Table 7. The comparison between the VMD+LSTM model in Ding and Qin (2020) and my model.

Models	RMSE	MAE	MAPE	MLSE
SH600777				
VMD-LSTM (sum)	1.96	1.97	100	1.18
VMD-LSTM (cascade)	0.5	0.5	100	0.16
SZ300161				
VMD-LSTM (sum)	2.06	2.06	100	1.25
VMD-LSTM (cascade)	0.49	0.48	100	0.16
SZ002323				
VMD-LSTM (sum)	1.83	1.83	100	1.08
VMD-LSTM (cascade)	0.48	0.48	100	0.15
SH000001				
VMD-LSTM (sum)	2.21	2.2	100	1.36
VMD-LSTM (cascade)	0.48	0.48	100	0.15
Average value				
VMD-LSTM (sum)	2.045	2.047	100	1.24
VMD-LSTM (cascade)	0.48	0.494	100	0.16

model at different time levels. And I changed the length of the index due to the impact of time changes on the forecast horizon. With finer time division, the measurement interval will be enlarged to provide a more convincing evaluation index. The corresponding relationship of the measurement interval is as follows: 1 day = 30, 1 h = 120, 30 min = 240, 10 min = 720. The meaning of these numbers is that for the day-level forecast we forecast 30 data points into the future, and since there are four trading hours in a day, for the hour

Table 8. Performance evaluation of cascade LSTM of different time levels.

Stock code	Time division	RMSE	MAE	MAPE	MSLE
SH600777	1 day	0.12	0.118	5.47	0.002
	1 h	0.03	0.02	1.02	0.00008
	30 min	0.02	0.02	0.91	0.00006
	10 min	0.02	0.01	0.65	0.00002
SZ300161	1 day	0.85	0.62	4.17	0.003
	1 h	0.38	0.28	1.84	0.0006
	30 min	0.27	0.19	1.25	0.0002
	10 min	0.14	0.09	0.61	0.00007
SZ002323	1 day	0.89	0.85	31.95	0.07
	1 h	0.08	0.07	3.82	0.007
	30 min	0.20	0.19	9.69	0.005
	10 min	0.11	0.10	5.55	0.002
SH000001	1 day	56.01	44.93	1.56	0.0004
	1 h	27.18	18.66	0.63	0.00008
	30 min	18.90	12.93	0.44	0.00004
	10 min	9.71	6.51	0.22	0.00001
Average value	1 day	0.70	0.62	16.53	0.006
	1 h	0.27	0.21	2.86	0.003
	30 min	0.22	0.17	4.32	0.002
	10 min	0.10	0.08	1.35	0.0001

level we forecast 4 times the day level. 30-min level and 10-min level are also multiplied by a multiple each. They are uniformly measured at the day level and are therefore comparable.

Table 8 shows that the finer the time partition, the more accurate the model performs. And the stock SH600777 performs better than the SZ002323. The difference between the two stocks is that they have the lowest volatility and the highest volatility respectively. The conclusion shown in tables is still consistent with the increasing Figure 3 above. The larger data values for the SH000001 make the RMSE and MAE values larger as well, but the excellent performance of MSLE and MAPE proves that the model is still applicable to it.

5. Conclusion

In this paper, a LASOO-based feature selection approach is combined with the cascaded LSTM (Ca-LSTM) network. Our proposed Ca-LSTM is based on the idea of reorganising data features and enhancing the impact of single features by cascading. Through a comprehensive comparison with the pure with other commonly used models on four datasets with dissimilar properties and thus guaranteed fairness, containing LSTM, XGboost, SVR, ARIMA, PSO-SRNN and the VMD-LSTM model, we have demonstrated the superior forecasting ability of our proposed model, especially on minute level data sets. Additionally, across a dataset of 30 selected stocks, we also find the shortcomings of the proposed model.

The advantages and disadvantages of the model are shown as follows: Firstly, the shorter the time period of the data used for prediction, the better the model's performance. Secondly, the model performed better for stocks with lower volatility. Thirdly, if the LASSO weight value is greater than zero, the effect is optimal. Furthermore, when the cascaded LSTM network is applied in the components which generated by the VMD network, it performs better than the method in the reference paper.

Our model not only contributes to the stock price prediction model in terms of single feature enhancement, but also provides a feasible direction, which is to use a better algorithm

to select features, and then input them into the network model after feature enhancement. In the future, new model can be trained from our proposed model, and we plan to explore the use of more algorithms (for example, use GA for parameter adjustment, use ATTENTION to explore the inner structure) to develop our model and investigate the applicability of our proposed approach to other financial time series data.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Data availability statement

The datasets used in this study are not publicly available but can be obtained from the corresponding author upon reasonable request.

ORCID

Zhi Cao  <http://orcid.org/0000-0002-3232-5200>

References

- Alam, W., Ray, M., Kumar, R. R., Sinha, K., Rathod, S., & Singh, K. (2018). Improved ARIMAX modal based on ANN and SVM approaches for forecasting rice yield using weather variables. *The Indian Journal of Agricultural Sciences*, 88(12), 1909–1913. <https://doi.org/10.56093/ijas.v88i12.85446>
- Bali, T. G., & Hovakimian, A. (2009). Volatility spreads and expected stock returns. *Management Science*, 55(11), 1797–1812. <https://doi.org/10.1287/mnsc.1090.1063>
- Bas, E., Egrioglu, E., & Kolenen, E. (2022). Training simple recurrent deep artificial neural network for forecasting using particle swarm optimization. *Granular Computing*, 7(2), 411–420. <https://doi.org/10.1007/s41066-021-00274-2>
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3), 307–327. [https://doi.org/10.1016/0304-4076\(86\)90063-1](https://doi.org/10.1016/0304-4076(86)90063-1)
- Bontempi, G., Ben Taieb, S., & Borgne, Y.-A. L. (2012). Machine learning strategies for time series forecasting. In *European business intelligence summer school* (pp. 62–77). Springer.
- Chang, P.-C., Liu, C.-H., Lin, J.-L., Fan, C.-Y., & Ng, C. S. (2009). A neural network with a case based dynamic window for stock trading prediction. *Expert Systems with Applications*, 36(3), 6889–6898. <https://doi.org/10.1016/j.eswa.2008.08.077>
- Chen, K., Le, C., Zhong, S., Guo, L., & Xu, G. (2022). NNNPE: Non-neighbourhood and neighbourhood preserving embedding. *Connection Science*, 34(1), 2615–2629. <https://doi.org/10.1080/09540091.2022.2133082>
- Chen, S., & Ge, L. (2019). Exploring the attention mechanism in LSTM-based Hong Kong stock price movement prediction. *Quantitative Finance*, 19(9), 1507–1515. <https://doi.org/10.1080/14697688.2019.1622287>
- Coelho, F., Costa, M., Verleysen, M., & Braga, A. P. (2020). Lasso multi-objective learning algorithm for feature selection. *Soft Computing*, 24(17), 13209–13217. <https://doi.org/10.1007/s00500-020-04734-w>
- Ding, G., & Qin, L. (2020). Study on the prediction of stock price based on the associated network model of LSTM. *International Journal of Machine Learning and Cybernetics*, 11(6), 1307–1317. <https://doi.org/10.1007/s13042-019-01041-1>
- Fu, L., Ding, X., & Ding, Y. (2022). Ensemble empirical mode decomposition-based preprocessing method with multi-LSTM for time series forecasting: A case study for hog prices. *Connection Science*, 34(1), 2177–2200. <https://doi.org/10.1080/09540091.2022.2111404>
- Gao, Y., Wang, R., & Zhou, E. (2021). Stock prediction based on optimized LSTM and GRU models. *Scientific Programming*, 2021(4), 1–8. <https://doi.org/10.1155/2021/4055281>

- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hollis, T., Viscardi, A., & Yi, S. E. (2018). A comparison of LSTMs and attention mechanisms for forecasting financial time series. Preprint [arXiv:1812.07699](https://arxiv.org/abs/1812.07699).
- Hoseinzadeh, S., Sohani, A., & Ashrafi, T. G. (2022). An artificial intelligence-based prediction way to describe flowing a Newtonian liquid/gas on a permeable flat surface. *Journal of Thermal Analysis and Calorimetry*, 147(6), 4403–4409. <https://doi.org/10.1007/s10973-021-10811-5>
- Kazem, A., Sharifi, E., Hussain, F. K., Saberi, M., & Hussain, O. K. (2013). Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Applied Soft Computing*, 13(2), 947–958. <https://doi.org/10.1016/j.asoc.2012.09.024>
- Kumar, K., Haider, M., & Uddin, T. (2021). Enhanced prediction of intra-day stock market using metaheuristic optimization on RNN–LSTM network. *New Generation Computing*, 39(1), 231–272. <https://doi.org/10.1007/s00354-020-00104-0>
- Li, J., Zhou, T., & Hu, X. (2022). Prediction algorithm of stock holdings of Hong Kong-funded institutions based on optimized PCA-LSTM model. *International Journal of Innovative Computing, Information and Control*, 18(3), 999–1008.
- Li, Q., Tan, J., Wang, J., & Chen, H. (2020). A multimodal event-driven LSTM model for stock prediction using online news. *IEEE Transactions on Knowledge and Data Engineering*, 33(10), 3323–3337. <https://doi.org/10.1109/TKDE.2020.2968894>
- Li, S., Tian, Z., & Li, Y. (2023). Residual long short-term memory network with multi-source and multi-frequency information fusion: An application to China's stock market. *Information Sciences*, 622, 133–147. <https://doi.org/10.1016/j.ins.2022.11.136>
- Li, Y., Dai, H.-N., & Zheng, Z. (2022). Selective transfer learning with adversarial training for stock movement prediction. *Connection Science*, 34(1), 492–510. <https://doi.org/10.1080/09540091.2021.2021143>
- Mahmoudan, A., Esmailion, F., Hoseinzadeh, S., Soltani, M., Ahmadi, P., & Rosen, M. (2022). A geothermal and solar-based multigeneration system integrated with a TEG unit: Development, 3E analyses, and multi-objective optimization. *Applied Energy*, 308, 118399. <https://doi.org/10.1016/j.apenergy.2021.118399>
- Mahmoudan, A., Samadof, P., Hosseinzadeh, S., & Garcia, D. A. (2021). A multigeneration cascade system using ground-source energy with cold recovery: 3E analyses and multi-objective optimization. *Energy*, 233, 121185. <https://doi.org/10.1016/j.energy.2021.121185>
- Matias, J. M., & Reboredo, J. C. (2012). Forecasting performance of nonlinear models for intraday stock returns. *Journal of Forecasting*, 31(2), 172–188. <https://doi.org/10.1002/for.v31.2>
- Niu, H., Xu, K., & Wang, W. (2020). A hybrid stock price index forecasting model based on variational mode decomposition and LSTM network. *Applied Intelligence*, 50(12), 4296–4309. <https://doi.org/10.1007/s10489-020-01814-0>
- Piramuthu, S. (2004). Evaluating feature selection methods for learning in data mining applications. *European Journal of Operational Research*, 156(2), 483–494. [https://doi.org/10.1016/S0377-2217\(02\)00911-6](https://doi.org/10.1016/S0377-2217(02)00911-6)
- Qiu, J., Wang, B., & Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PLoS One*, 15(1), e0227222. <https://doi.org/10.1371/journal.pone.0227222>
- Rounaghi, M. M., & Zadeh, F. N. (2016). Investigation of market efficiency and financial stability between S&P 500 and london stock exchange: Monthly and yearly forecasting of time series stock returns using ARMA model. *Physica A: Statistical Mechanics and Its Applications*, 456, 10–21. <https://doi.org/10.1016/j.physa.2016.03.006>
- Sohani, A., Hoseinzadeh, S., Samiezadeh, S., & Verhaert, I. (2021). Machine learning prediction approach for dynamic performance modeling of an enhanced solar still desalination system. *Journal of Thermal Analysis and Calorimetry*, 147, 3919–3930. <https://doi.org/10.1007/s10973-021-10744-z>

- Sohani, A., Pedram, M. Z., Berenjkari, K., Sayyaadi, H., Hoseinzadeh, S., Kariman, H., & Assad, M. E. H. (2021). Techno-energy-enviro-economic multi-objective optimization to determine the best operating conditions for preparing toluene in an industrial setup. *Journal of Cleaner Production*, 313, 127887. <https://doi.org/10.1016/j.jclepro.2021.127887>
- Song, Y., Lee, J. W., & Lee, J. (2019). A study on novel filtering and relationship between input-features and target-vectors in a deep learning model for stock price prediction. *Applied Intelligence*, 49(3), 897–911. <https://doi.org/10.1007/s10489-018-1308-x>
- Soni, P., Tewari, Y., & Krishnan, D. (2022). Machine learning approaches in stock price prediction: A systematic review. *Journal of Physics: Conference Series*, 2161, 012065.
- Su, Z., Xie, H., & Han, L. (2021). Multi-factor RFG-LSTM algorithm for stock sequence predicting. *Computational Economics*, 57(4), 1041–1058. <https://doi.org/10.1007/s10614-020-10008-2>
- Sun, L., Xu, W., & Liu, J. (2021). Two-channel attention mechanism fusion model of stock price prediction based on CNN-LSTM. *Transactions on Asian and Low-Resource Language Information Processing*, 20(5), 1–12. <https://doi.org/10.1145/3453693>
- Tang, M., Chen, W., & Yang, W. (2022). Anomaly detection of industrial state quantity time-series data based on correlation and long short-term memory. *Connection Science*, 34(1), 2048–2065. <https://doi.org/10.1080/09540091.2022.2092594>
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267–288.
- Wang, J., Cheng, Q., & Dong, Y. (2022). An XGBoost-based multivariate deep learning framework for stock index futures price forecasting. *Kybernetes*, 52(10), 4158–4177.
- Wang, J., Cui, Q., Sun, X., & He, M. (2022). Asian stock markets closing index forecast based on secondary decomposition, multi-factor analysis and attention-based LSTM model. *Engineering Applications of Artificial Intelligence*, 113, 104908. <https://doi.org/10.1016/j.engappai.2022.104908>
- Wang, J., & Zhu, S. (2022). A multi-factor two-stage deep integration model for stock price prediction based on intelligent optimization and feature clustering. *Artificial Intelligence Review*, 56, 7237–<https://doi.org/10.1007/s10462-022-10352-9>
- Wang, J.-Z., Wang, J.-J., Zhang, Z.-G., & Guo, S.-P. (2011). Forecasting stock indices with back propagation neural network. *Expert Systems with Applications*, 38(11), 14346–14355. <https://doi.org/10.1016/j.eswa.2011.04.222>
- Wu, S., Liu, Y., Zou, Z., & Weng, T.-H. (2022). S_i_lstm: Stock price prediction based on multiple data sources and sentiment analysis. *Connection Science*, 34(1), 44–62. <https://doi.org/10.1080/09540091.2021.1940101>
- Xia, Y., Liu, Y., & Chen, Z. (2013). Support vector regression for prediction of stock trend. In *2013 6th International conference on information management, innovation management and industrial engineering* (Vol. 2, pp. 123–126). IEEE.
- Yu, Y., & Kim, Y.-J. (2019). Two-dimensional attention-based LSTM model for stock index prediction. *Journal of Information Processing Systems*, 15(5), 1231–1242.
- Yu, Z., Qin, L., Chen, Y., & Parmar, M. D. (2020). Stock price forecasting based on LLE-BP neural network model. *Physica A: Statistical Mechanics and Its Applications*, 553, 124197. <https://doi.org/10.1016/j.physa.2020.124197>
- Yujun, Y., Yimei, Y., & Wang, Z. (2021). Research on a hybrid prediction model for stock price based on long short-term memory and variational mode decomposition. *Soft Computing*, 25(21), 13513–13531. <https://doi.org/10.1007/s00500-021-06122-4>
- Zahedi, J., & Rounaghi, M. M. (2015). Application of artificial neural network models and principal component analysis method in predicting stock prices on Tehran stock exchange. *Physica A: Statistical Mechanics and Its Applications*, 438, 178–187. <https://doi.org/10.1016/j.physa.2015.06.033>
- Zhao, Y., & Yang, G. (2023). Deep learning-based integrated framework for stock price movement prediction. *Applied Soft Computing*, 133, 109921. <https://doi.org/10.1016/j.asoc.2022.109921>