Stock Prediction Based on LSTM

Min Li[1,a], Weize Liao[2,3,b], Jianrong Huang[2,c*], Jiebo Jiang[4,d*]

{125964643@qq.com[a], 20074020@qq.com[b], 1375480613@qq.com[c*], 359128664@qq.com[d*]}

Guangxi Xijiang Development Investment Group Co., Ltd. Lock Operation Management Branch,
543100, Wuzhou, China[1]
Wuzhou University, 543002, Wuzhou,China[2]
Guangxi Key Laboratory of Machine Vision and Intelligent Control, 543002 China[3*]
Cangwu County Information Center, Wuzhou, China[4*].

Abstract. Stock market forecasting based on Long Short-Term Memory networks (LSTMs) is a prevalent technique within the realms of Natural Language Processing and time-series analysis. Predicting stock market trends is widely acknowledged to be a highly intricate task, attributed to the multifaceted influences ranging from economic factors, policy changes, to market sentiment. The LSTM's superior capability in handling time-series data has thus made it a popular choice for such forecasting endeavors. This paper selects two stocks, Vanadium Titanium Shares (SZ.000629) and Crystal Technology (SH.603005), and applies both single variable LSTM(SV-LSTM) and multivariate LSTM(MV-LSTM) models to forecast the opening price, the lowest price, and the previous closing price. The results indicate that the MV-LSTM yields more accurate predictions and exhibits relatively smaller errors than its SV-LSTM counterpart.

Keywords: Stock; LSTM; Prediction; Multivariate

1 Introduction

Economics is the cornerstone of a nation, and the financial sector plays a pivotal role in the development of the national economy, becoming an indispensable part of the modern economic framework. The abundance of research on stocks, both domestically and internationally, highlights the significant importance of stock analysis. In stock prediction research, Luo Yukun (2022) improved the LSTM model with Convolutional Neural Networks (CNN) and combined it with the GM (1,1) grey model to construct a hybrid model for stock price forecasting studies[1]. Huang Yutang (2022) employed seven types of time series neural network models—RNN, LSTM, GRU, Bi-LSTM, Bi-GRU, ConvLSTM, TCN—along with five indicators of stock fundamentals to train and predict the closing prices of the constituents of the SSE 50, demonstrating varying effects of different models on different stocks[2]. Li Kaimin (2022) used the Fin-BERT sentiment analysis model for news and commentary analysis, adopted a semantic rule model to acquire text sentiment, and constructed a daily investor sentiment index using the half-life weighting method. Subsequently, Li validated the relationship between the sentiment index and stock returns using the Granger causality test and predicted returns with an LSTM model using the sentiment index as an input factor, resulting in good predictive performance[3]. Geng Sujuan computed sentiment scores with SnowNLP and jieba segmentation to obtain a sentiment index, used a bivariate VAR model to map the relationship between online sentiment and stock returns, and finally created univariate

MSIEID 2023, December 08-10, Guangzhou, People's Republic of China

DOI 10.4108/eai.8-12-2023.2344733

and multivariate LSTM models incorporating network sentiment for stock return predictions, with empirical results favoring the multivariate prediction model integrating sentiment[4]. Internationally, researchers such as Kanwal Anika and others have proposed a hybrid deep learning-based prediction model combining Bi-directional Cuda Deep Neural Network Long Short-Term Memory (BiCuDNNLSTM) with a set of Convolutional Neural Networks (CNN), which has improved prediction accuracy[5]. Swathi introduced a Teaching-Learning-Based Optimization (TLBO) model which used the LSTM model for sentiment analysis of Twitter data, determining learning rates with the Adam optimizer, and optimizing the output units of the LSTM model with the TLBO, yielding promising results[6].

2 DataSet

2.1 Data Collection

The acquisition of data is crucial for our subsequent analysis. To this end, considering factors such as activity level and market capitalization, we selected historical data from January 2017 to August 2022 for SZ.000629 and SH.603005 Stocks. This data was obtained via Tushare, a Python library for crawling historical trading data, which includes open price, closing price, highest price, and lowest price, among others. To utilize Tushare for retrieving historical trading data of stocks, one must first register an account with the Tushare big data community. Successful registration provides a TOKEN value that is required for accessing stock data. The retrieval process involves using the t1.get_hist_data method, inputting the stock code and the start and end dates to fetch the data, which can then be exported to Excel for ease of further analysis. As indicated in Table 1, "Historical Trading Data for Vanadium Titanium Shares," we have obtained information such as the stock's opening price, highest price, and closing price. We have chosen the closing price as the variable for predicting the stock's closing price.

Table 1. Historical Market Data of SZ.000629.

date	open	high	close	low	volume	price change	p_change
2022/8/5	6.46	6.52	6.44	6.28	2730741.50	-0.01	-0.15
2022/8/4	6.40	6.52	6.45	6.27	3194291.25	0.05	0.78
2022/8/3	6.64	6.91	6.40	6.38	5023046.50	-0.26	-3.90
2022/8/2	7.11	7.12	6.66	6.54	5415491.00	-0.61	-8.39
2022/8/1	7.31	7.39	7.27	7.10	3574441.50	-0.12	-1.62
2022/7/29	7.11	7.67	7.39	6.97	6290002.50	0.25	3.50
2022/7/28	7.25	7.49	7.14	6.94	4602362.50	-0.04	-0.56
2022/7/27	7.00	7.29	7.18	7.00	4233641.50	0.17	2.42
2022/7/26	6.90	7.23	7.01	6.76	4344194.50	-0.05	-0.71
2022/7/25	7.40	7.40	7.06	6.83	6088807.00	-0.42	-5.62
2022/7/22	7.26	7.75	7.48	7.13	7310199.00	0.01	0.13
2022/7/21	7.07	7.87	7.47	6.94	11118755.00	0.32	4.48
2022/7/20	6.30	7.15	7.15	6.21	9649823.00	0.65	10.00

3 Model and Method

LSTM represent a specialized subset of RNNs. Traditional RNNs, during training, tend to encounter issues with gradient explosion or vanishing gradients as the length of training and the number of network layers increase. This results in an inability to process and leverage information from longer data sequences. To address this challenge, LSTMs are engineered with a sophisticated gate structure that regulates the removal or addition of information to the cell state. These gates function as selective information pathways, with LSTMs primarily comprising three types of gates—the forget gate, the input gate, and the output gate—alongside a memory cell. These gates are instrumental in protecting and controlling the cell state[7].

3.1 SV-LSTM

In this section, we employ a univariate time series to train an LSTM for the purpose of predicting stock closing prices. The detailed design rationale for the SV-LSTM (Single Variable Long Short-Term Memory) forecasting of stock closing prices is as follows:

To constrain the data within a specific bandwidth and to mitigate the adverse effect of outlier values on the time series, which could lead to a non-standard distribution or an excessive standard deviation of input series—potentially slowing the learning and convergence speed of the neural network and hindering learning efficiency—we utilize the z-score normalization method for data standardization preprocessing. This transforms the corresponding x to a normalized y. Subsequently, we define the training set and iteratively calculate the number of training samples[8].

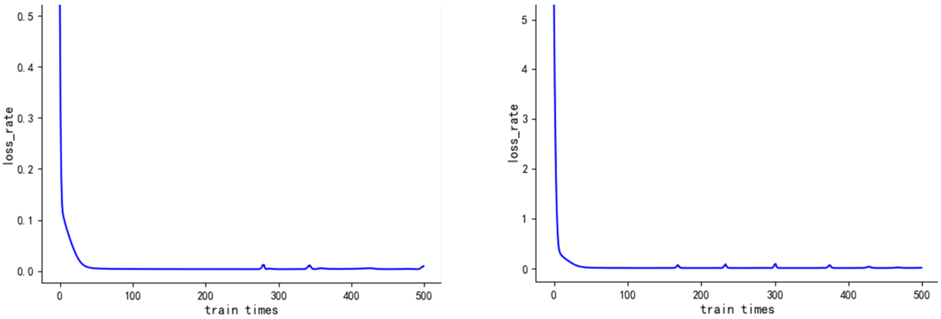
Following this, we define the input layer, weights, biases, and the number of batch inputs to the neural network. For matrix multiplication, we convert batch data into a two-dimensional array, then transform the outcome of linear computation back into three dimensions to serve as input to the cell. We define the input dimension of the cell and for each training iteration, an initial state is required. The output from the last cell is taken as the input to the output layer, upon which further computation is conducted.

We define the loss function and employ the Adam optimizer for the training algorithm, which ensures a determined range for the learning rate in each iteration, facilitating stable parameter adjustment. We also set the model to save iteratively—repeating training 500 times, with the model saved every 10 steps. We then plot the relationship between loss and the number of training iterations.

For the prediction model, we load the last saved (existing) model, restore variables, and save the computed prediction results in a newly defined list. The results are converted into one-dimensional data and inverted back to pre-normalization values, with the final step being the plotting of the predicted prices versus actual price movements.

During this process, it is necessary to adjust the time step length, number of training samples per batch, and the number of training iterations. Employing the control variable method, we alter one parameter while holding others constant and record the MAE value. For SZ.000629, we set the time step length to 3, batch size to 64, number of training iterations to 500, and the learning rate to 0.001, with dimensionality of output and input layers set to 1, all corresponding to closing prices. For SH.603005, a time step length of 4 yielded better convergence of the loss function; all other parameters remained consistent with SZ.000629.

Figure 1 portrays the gradient descent of the loss function during the training process of the SV-LSTM model for both stocks. In the case of SZ.000629, when the model training iteration reaches 500, the loss progressively diminishes and fluctuates negligibly around zero, implying a stable loss. In SH.603005, after 400 of the 500 training iterations, no significant change is observed, and there are only minor fluctuations around zero. Thus, the model is convergent for both stocks and can be utilized for predictive purposes.



(a) SZ.000629 SV-LSTM

(b) SH.603005 SV-LSTM

Fig. 1. SV-LSTM Loss Function

3.2 MV-LSTM

In this section, we conduct stock training and prediction using a multivariate time series composition, where 'multivariate' refers to the inclusion of multi-dimensional time series data. The time series is constructed using five variables: opening price, lowest price, highest price, previous closing price, and percentage change, which are utilized to train and predict the final closing price. The design rationale is outlined as follows:

Similar to the SV-LSTM, the MV-LSTM also utilizes the z-score normalization method for data standardization pre-processing to rectify issues related to non-standard time series distributions or large standard deviations, thus enhancing the speed of learning and convergence. Data is imported, and the required columns are read, with the first 1,000 records designated as the training set. After standardization, the data are converted into a list assigned to x with a dimensionality of 5, and y stores the label—with np.newaxis increasing the dimension for predicted value (closing price). Data beyond the first 1,000 records are used as the test set and undergo similar standardization. Weights and biases are initialized randomly, and both the batch size and the time step length for input into the neural network are defined. As with the univariate model, the X (input batch size) data must be transformed into a two-dimensional array, which after linear calculation is reverted back to three dimensions to serve as the input to the cell, and the input dimension of the cell is defined. An initial state is required for each training cycle, and the output from the last cell is used as the input to the output layer, which is then calculated. The training model defines the input layer and loss function; the training set undergoes training using the Adam optimization algorithm to ensure the learning rate is within a defined range for each iteration, stabilizing the parameters. The model is set to save after a defined number of training iterations—500 in total with a save every 200 iterations—followed by plotting a graph of loss vs. training iterations. For prediction, the test data is obtained, the last saved model is read, variables are restored, and the computed predictions are saved to a newly defined list. Results are transformed into one-dimensional data and reverted to pre-standardization values before plotting the predicted versus actual price fluctuations[9].

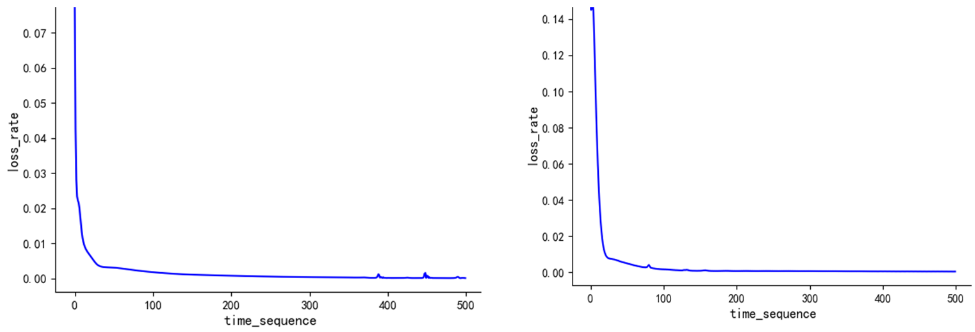
Adjustments to the time step length, batch size per training iteration, and the number of training iterations are also necessary. For the prediction of multivariate stock closing prices for SZ.000629 and SH.603005 stocks, the parameters used are as follows: a time step length of 3, batch size of 64, number of training iterations is 500, and a learning rate of 0.006, with the number of output neurons set to 1 and input neurons set to 5, as shown in Table 2.

Table 2. SZ.000629 MV-LSTM hyperparameter settings

Hyperparameter	Value Setting	Summary
time_step	3	Time interval days
rnn_unit	10	hidden layer units
input_size	5	The number of input neurons
output_size	1	Number of output neurons (predicted value)
lr	0.0006	Learning rate

In the process of acquiring training and test datasets, it is requisite to standardize the data by calculating the mean and standard deviation for each column, and by generating random initial values for weights (w) and biases (b).

The descent of the MV-LSTM loss function during the training process for the two stocks is depicted in Figure 2. The horizontal axis denotes the number of training iterations, while the vertical axis displays the value of the loss function. The discrepancy between the predicted closing price for individual training samples and the actual values gradually diminishes towards zero, reaching a state of stability. Consequently, the model demonstrates convergence for both stocks and is deemed suitable for predictive application.



(a) SZ.000629 MV-LSTM

(b) SH.603005 MV-LSTM

Fig. 2. MV-LSTM Loss Function

3.3 Algorithm

The simplified pseudo-code for MV-LSTM Algorithm is as follow.The SV-LSTM algorithm is similar, with the only difference being that the input is 1-dimensional.

```
1. Initialize LSTM with input_size = 5 (open, low, high, previous close, % change)
2. Prepare time series data into sequences of length 'time_step'
3. Define LSTM network with 'rnn_unit' LSTM cells
4. Forward pass: input sequences through LSTM to get output
5. Calculate loss using actual closing prices and predicted prices from LSTM
6. Backpropagate errors to adjust weights
7. Update the model using an optimizer with set 'learning_rate'
8. Repeat steps 4-7 for 'epochs' iterations in batches of size 'batch_size'
9. Use trained LSTM to predict closing price given new input sequence
10. Evaluate model performance on test data
```

Note that actual code would involve specific function calls and more settings, especially for steps involving forward pass, loss calculation, backpropagation, and the update step. This pseudo-code is a high-level abstraction.

3.4 Evaluation Metrics

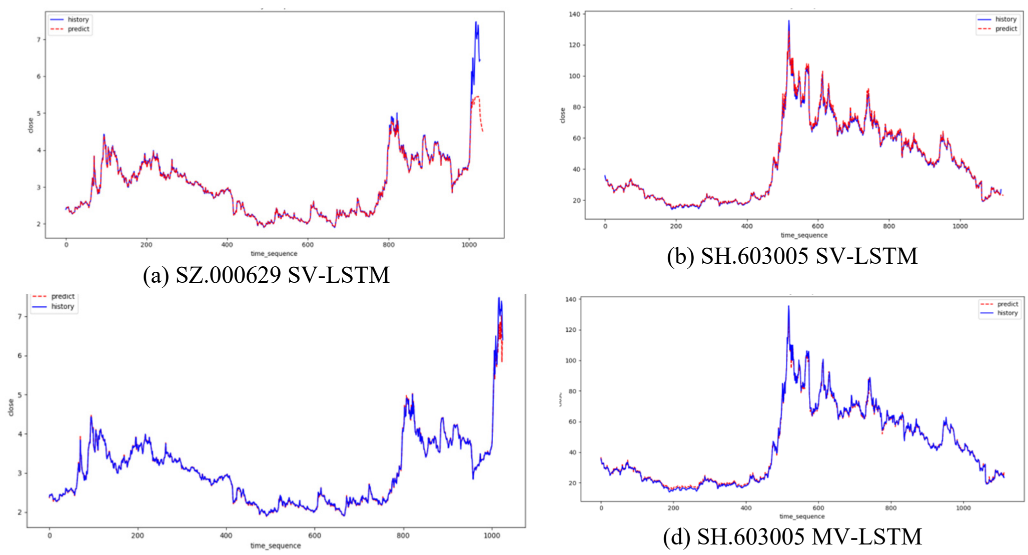
The prediction of stock closing prices plays a significant role in providing reference values for investors and in mitigating financial risks. This chapter delineates the application of the LSTM model to forecast stock closing prices, with the predictive performance being evaluated using the Mean Absolute Error (MAE)[10]. The MAE is expressed by the following formula:

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|f(x_i) - y_i| \tag{1}$$

In formula 1, $f(x_i)$ represents the predicted values, y_i denotes the actual values, and m is the sample size. The MAE provides an accurate reflection of the magnitude of the actual predictive error. The merit of the predictive outcomes is adjudicated based on the MAE value—with a smaller MAE indicating more accurate predictions.

4 Results and Discussions

4.1 Results



(a) SZ.000629 SV-LSTM

(b) SH.603005 SV-LSTM

(c) SZ.000629 MV-LSTM

(d) SH.603005 MV-LSTM

Fig 3. Visualization of Prediction Results

Figure 3 graphically illustrates our predictive analytics. The abscissa (horizontal axis) marks the progression of time, and the ordinate (vertical axis) emblematizes the closing price of stocks. The solid blue line delineates the historical trajectory of closing prices. In contrast, the red dashed line represents the estimated closing price curve, procured through advanced predictive models. Utilizing the SV-LSTM and the MV-LSTM models, we produced forecasts for two stocks which are graphically represented.

With particular attention to Figure 3(a), it presents the SV-LSTM model's predictive results for stock SZ.000629. Here, the observed Mean Absolute Error (MAE) of 0.08 signifies a palpable variance from the historical values. This error quantifies the average magnitude of the discrepancies between predicted and actual values, which are calculated using the formula 1.

Figure 3(b) delineates the univariate prediction for stock SH.603005. Compared to Figure 3(a), the predictive line hews closer to the historical trend, delivering a slimmer error margin evidenced by an MAE of 1.453.In juxtaposition, Figures 3(c) and (d) showcase the more precise forecast results of the MV-LSTM model. The MAEs of 0.0249 and 0.807 for the corresponding stocks underscore the enhanced accuracy of multivariate over univariate predictions.

4.2 Discussions

An evaluation of the presented data against the backdrop of MAE indicator, as aggregated in Table 3, substantiates the heightened precision of the MV-LSTM model over the SV-LSTM when forecasting stock prices. The empirical assessment, emulated via Figure 3, manifests the comparative advantage of multivariate forecasts. The convergence of predicted and actual values in multivariate scenarios is demonstrably tighter, thereby ratifying the inference that multivariate predictions are invariably more robust and reliable vis-à-vis their univariate analogs.

Table 3. Comparative of MAE Values in SV-LSTM and MV-LSTM Forecasts

Models	SZ.000629 MAE	SH.603005 MAE
SV-LSTM	0.0800	1.453
MV-LSTM	0.0249	0.807

In this table, it is explicit that MV-LSTM model cuts a more accurate and reliable forecasting methodology, as evinced by the MAE metrics. These models are formulated by programming algorithms designed to capture the nuanced interactions between multiple variables influencing stock movements, providing a robust scaffold for our forecasts.

5 Conclusions

In conclusion, the application of LSTM in the time series analysis of stock market prediction affirms its validity as a robust method in addressing the complexity of forecasting stock prices. The inherent volatility of stock prices, influenced by a multitude of economic factors, policy shifts, and market sentiments, poses a significant challenge that LSTM networks are well-equipped to handle due to their exceptional ability to process temporal data sequences. This study specifically examined the efficacy of both SV-LSTM and MV-LSTM models by forecasting the opening price, lowest price, and previous closing price of two selected stocks, SZ.000629 and SH.603005. The comparative results indicate a superior performance of the multivariate LSTM model over its univariate counterpart, with reduced error margins, thereby demonstrating the potential advantages of incorporating multiple variables into the forecasting model. The findings underscore the importance of selecting appropriate model complexities that can capture the dynamics of the stock market more effectively, paving the way for more accurate and reliable predictive analyses in the financial domain.

References

[1] Luo, Y. K.: Study on stock prediction based on improved LSTM and grey model [D]. Anqing Normal University, 2022. DOI: 10.27761/d.cnki.gaqsf.2022.000256.

[2] Huang, Y. T.: Comparative Study on Stock Prediction Based on Neural Networks and Their Ensemble Machine Learning [D]. Minzu University of China, 2022. DOI: 10.27667/d.cnki.gzymu.2022.000272.

[3] Kanwal, A., Lau, M. F., Ng, S. P. H., et al.: BiCuDNNLSTM-1dCNN — A hybrid deep learning-based predictive model for stock price prediction [J]. Expert Systems With Applications, 2022, 202.

[4] Swathi, T., Kasiviswanath, N., Rao, A. A., et al.: An optimal deep learning-based LSTM for stock price prediction using twitter sentiment analysis [J]. Applied Intelligence, 2022: 1-14.

[5] Gebka, B.: The Non-linear and Linear Impact of Investor Sentiment on Stock Returns: An Empirical Analysis of the US Market [Z]. 2013: 281-299.

[6] Oliveira, N., Cortez, P., Areal, N., et al.: The impact of microblogging data for stock market prediction: Using Twitter to predict returns, volatility, trading volume and survey sentiment indices [J]. Expert Systems With Applications, 2017, 73: 125-144.

[7] Xue, W.: Statistical Analysis and Application of SPSS, 5th Edition [M]. China Renmin University Press, 2017.06.

[8] Xue, X. Y., Long, J., Fang, Y. C.: Research on wireless network traffic prediction based on LSTM algorithm [J]. Yangtze Information Communication, 2021, 34(10): 4-6.

[9] Fang, J. L., Zuo, K., Huang, C., Liu, J., Li, S. G., Lu, K.: FD-LSTM: Fault diagnosis model based on large-scale system logs [J]. Computer Engineering and Science, 2021, 43(01): 33-41.

[10] Huang, H. J.; Li, B.: Stock price prediction based on the VMD-CSSA-LSTM combined model [J/OL]. Journal of Nanjing University of Information Science & Technology (Natural Science Edition), 1-13 [2023-11-26]. https://doi.org/10.13878/j.cnki.jnuist.20230903002..