ISSN 2616-5775 Vol. 5, Issue 12: 98-106, DOI: 10.25236/AJCIS.2022.051215

Research on stock prediction algorithm based on CNN and LSTM

Keren He[1,a], Qian Jiang[2,b,*]

[1]Information Center, Changzhou University, Changzhou, China
[2]School of Computer and Artificial Intelligence, Changzhou University, Changzhou, China
[a]hkrhjy@cczu.com.cn, [b]20081203040@smail.cczu.edu.cn
[*]Corresponding author: Qian Jiang

Abstract: *The traditional resolution to forecast stock trends accurately is based on time series models. However, traditional time series models simply cannot fit the irregular movements of the market due to their own limitations.This study combines the advantages of convolutional neural networks (CNN) and long short-term memory networks (LSTM) for improving the stock prediction accuracy, uses the convolution idea of CNN to construct feature extraction layer to extract features, and inputs the extracted features into LSTM, to study the temporal message of the features better. The model constructed in this study can capture local features and time series information of stock data.The performance indicators of the proposed model are shown to be outperformed by CNN and LSTM, and the method is effective in reducing forecasting errors.*

Keywords: *Stock prediction; Convolutional neural networks; Long short-term memory networks; Feature extraction*

1. Introduction

Many investors in the stock market constantly try to predict stocks to maximize their investment profits. However, predicting stocks is complex, and there have been many attempts to use various techniques[1][2][3][4]to predict stock prices for profit. The researchers[5]used econometric methods to process the data and design the model, the results indicated that the model is partially feasible for stock returns. The researchers[17] have improved the ARIMA model: genetic algorithm (GA) and particle swarm optimization particle swarm algorithm (PSO) are fused with ARIMA. The improved model is also applicable to forecast stock prices. A hybrid forecasting model[20] is proposed that uses multiple technical indicators to predict stock price movements. Experimental results showed that the model outperforms two RSTs and GA in terms of prediction accuracy.However, the econometric approach represents the stock count characteristic as a random procedure and uses the historical data for that characteristic to fit the stochastic process. This approach is suitable for linear and smooth time series and cannot fit nonlinear stock data, while neural networks (ANN) are not affected. Therefore, many scholars have studied stock price forecasting based on ANN models.[7] The researchers designed two models, one combining ANN with back propagation (BP) and the other incorporating ideas from genetic algorithms (GA) into ANN. After performing experimental comparisons, it was found that both models had high forecasting accuracy.[21] Researchers employed a genetic algorithm method to improve the discrete thresholding and connectivity weights. The experimental results of this method outperformed traditional neural networks. Long short-term memory (LSTM) is superior in learning time series. LSTM is designed to conquer the gradient disappearance problem in recurrent neural networks (RNN) by employing memory units and gates to learn the long-term correlation of time series data.LSTM [9]was used to fit and predict stock returns. A significantly improved accuracy of stock return prediction by LSTM from 14.3% level to 27.2% level, which experimentally proved that LSTM is an accurate stock prediction model. Researchers[8] used LSTM to forecast the future tendency of stock prices. It is indicated by the results that the proposed model is more precise than other models. Although LSTM performs well in modeling time series, it cannot extract features from stock data. CNN can learn and then withdraw the features.CNNs are excelling in computer vision, image processing and other deep learning techniques such as VGGNet[10], GoogLeNet[11]and ResNet[6]. A recent attempt is to apply stock data to CNN. A cnn-based framework is proposed[12], which can be used to extract features. The results indicated a significantly better prediction of the algorithm's performance. The researchers[13] put forward a new method based on a combination of wavelets and

CNNs, considering the features of multilayer perceptron (MLP), CNN and LSTM. The findings indicated that the model can predict financial time series and can also be trained on ordinary time series data.This paper analyses and models stock data by combining CNN and LSTM. This paper draws on the idea of CNN and designs its own layer for feature extraction.The feature extraction layer is built based on many unlabeled stock data, and the feature extraction layer's feature map is organized into sequence features, which are used as the input of LSTM to learn sequence correlation. This paper provides a time series analysis method for predicting stock trend volatility. By measuring how accurate various models are in predicting stock price movements, the model is demonstrated to be superior to a single model and provides insight into the most used time series models.

2. Principles of stock forecasting

Stock forecasting is a very competitive task with a variety of complex factors. It not only includes the serial characteristics of the stock, but also includes the financial data of the stock company and social opinion. Most models' input characteristics are based on primary data from historical markets; some models have been developed by mining financial news[14], social media text[15]and web browsing data[16].Stock forecasting principles: using mathematical and logical methods to generalize and summarise historical and current stock market movements to predict future changes in the stock market. Traditional stock forecasting methods based on linear analysis cannot accurately portray irregular stock changes, making it difficult to obtain highly accurate forecasting results. Neural networks have good non-linear processing techniques and can make accurate predictions by modeling the value and trend of stocks based on the intrinsic linkage of stock information. This study transforms stock data into an image structure and feeds into a feature extraction layer constructed using CNN ideas. The continuous output features are provided directly to the LSTM. Therefore, the trading model in this paper enables the LSTM to learn long-term dependence from higher-order serial elements.

3. Stock Forecasting Model

The structure of the model constructed in this paper involves two main parts: a feature extraction layer and LSTM. The next two sections describe the procedure of extracting high level stock feature sequences in the feature extraction layer and how to use LSTM to capture the long-term correlation of the feature sequences, respectively. The structure of the model is presented in Figure 1.

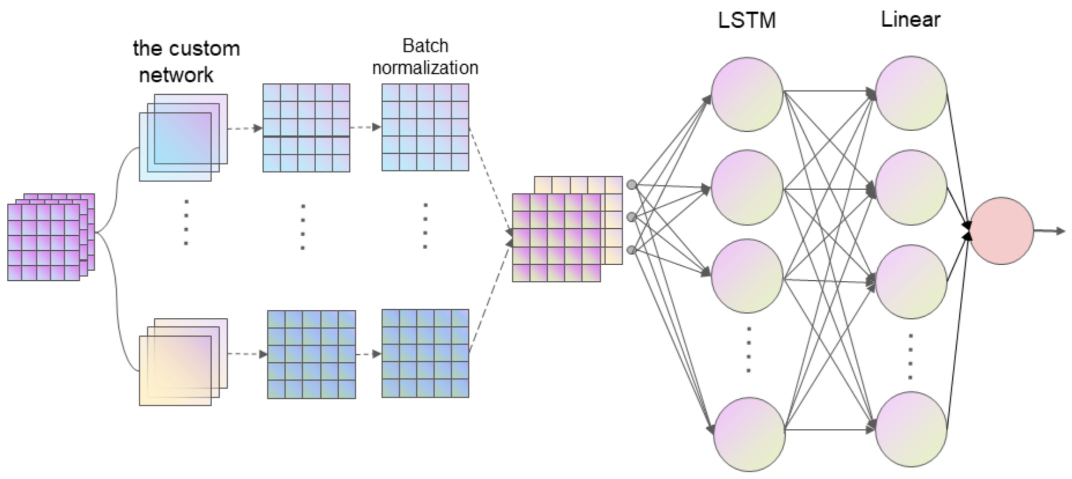

Figure 1: Model structure: consisting of raw data, feature extraction layer, LSTM

3.1. Extraction of stock data features

This paper combines multiple operator functions as a feature extraction layer, replacing the "convolution kernel" of the CNN to extract features from stock data. Table1 shows the custom network that the model has implemented.

Table 1: Six custom networks for the feature extraction layer

Name	Definition
$\mathrm{corr}(x_i, x_j, d)$	Correlation coefficients of x_i and x_j for the past d days
$\mathrm{cov}(x_i, x_j, d)$	Covariance of x_i and x_j for the past d days
$\mathrm{stddev}(x_i, d)$	Standard deviation of x_i for the past d days
$\mathrm{reccv}(x_i, d)$	The average of the x_i over the past d days divided by the standard deviation
$\mathrm{return}(x_i, d)$	$(x_i - \mathrm{delay}(x_i, d))/\mathrm{delay}(x_i, d) - 1, \mathrm{delay}(x_i, d)$ is the of x_i taken before d days
$\mathrm{wavg}(x_i, d)$	Weighted average of x_i over the past d days

Where $x_i \in X^t$ is a t-dimensional stock vector for the i feature in the stock data, $x_j \in X^t$ is denoted as a t-dimensional stock vector for the j feature in the stock data, and t is the number of days in the history of the selected stock. x_i^d denotes the data input to the custom network, and d is the number of days in the past.

The custom network will traverse the two-dimensional data in both the time and feature dimensions, setting the stride to 10. The computation of the custom network in the feature dimension reflects the difference from CNN convolution, where only local perception is possible. The custom network layer will traverse all types of stock data, with the computation regions not necessarily adjacent,and this avoids the data alignment problems associated with local perception in CNNs. Input x_i^d into the custom network to generate the feature mapping. Since $\mathrm{corr}(x_i, x_j, d)$and $\mathrm{cov}(x_i, x_j, d)$ require two different sets of data x_i, x_j, and there are C_9^2=36 selections of x_i, x_j in this paper, an inverted list of combinatorial number pairs is constructed for the custom network, which can effectively generate combinatorial number pairs and realize the functional requirements of the custom network. Take $\mathrm{cov}(x_i, x_j, d)$as an example. It is different to obtain the covariance matrix by arranging the data in the matrix by rows or columns. Rows arrange the data in this paper. Each row is a SAMPLE, and each column is a random variable. Enter the matrix with the following equation:

$$X_{m\times n} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} = \left[c_1, c_2, \ldots, c_n\right] \tag{1}$$

The covariance matrix is:

$$\mathrm{cov}\ Matrix = \frac{1}{m-1}\begin{pmatrix} \mathrm{cov}(c_1, c_1) & \cdots & \mathrm{cov}(c_1, c_n) \\ \vdots & \ddots & \vdots \\ \mathrm{cov}(c_n, c_1) & \cdots & \mathrm{cov}(c_n, c_n) \end{pmatrix} \tag{2}$$

The dimensionality of the covariance matrix is equal to the number of random variables, i.e., the dimensionality of each sample, so in this paper, it is $\frac{1}{m-1}$. The formula to calculate the covariance matrix is thus.

$$\mathrm{cov} = \frac{1}{m-1} X^T X \tag{3}$$

Based on the 36 different selection methods described above, 36 combinatorial pairs are designed. The pairs are used to slice the stock data to achieve computation in the feature dimension, with the custom network layer outputting a two-dimensional 'feature picture'. Each custom network is followed by a Batch Normalization (BN) layer. The feature extraction layer uses the BN layer with a weight-sharing strategy, treating each feature map as a single neuron. For example, the feature dimension of a layer is $[m, f, p, q]$where batch-num: m; dimension: f; feature dimension $[p, q]$.Thus, using the BN layer means finding the mean and variance of all neurons corresponding to the feature maps of all samples, and then standardizing all neurons. For n custom network models, the generated n features mapping can be reordered as:

$$W = [c_1, c_2, \cdots, c_n] \tag{4}$$

Here is the result of the nth custom network-generated feature mapping after BN. Subsequently, new sequential higher-order feature mappings are fed to the LSTM. LSTM is specified for sequential inputs, and the pooling layer would destroy this sequential organization, so pooling is not applied after the convolution operation. The specific steps for extracting the data features are shown in Figure 2. The feature formation of the feature extraction layer is illustrated in Figure 3.

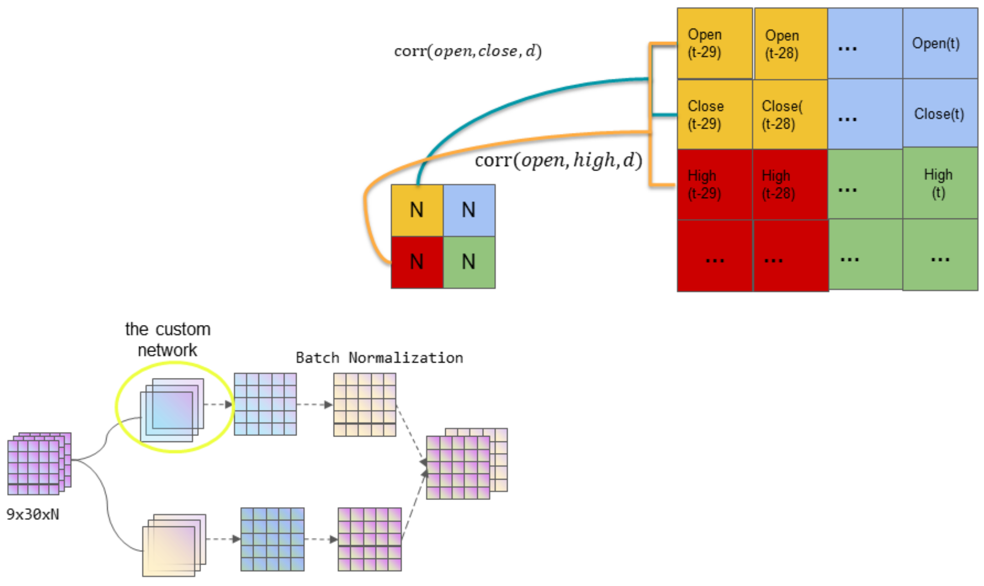

Figure 2: Working mechanism for extracting features from a custom network

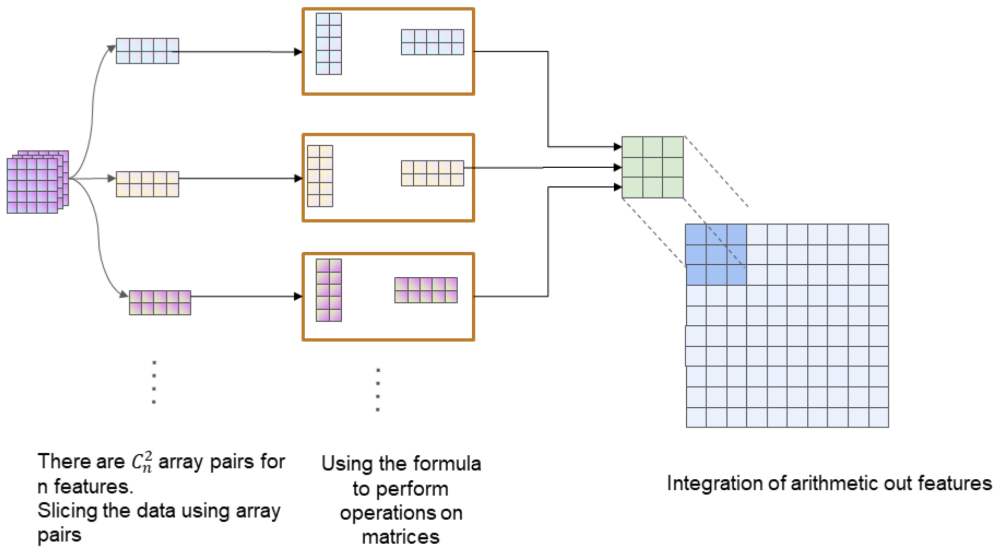
There are C_n^2 array pairs for n features.
Slicing the data using array pairs
Using the formula to perform operations on matrices
Integration of arithmetic out features

Figure 3: Feature extraction layer forming a featured image

3.2. Learning long-term time-series dependencies

When using RNN, it will assume that the current state is only related to a few states before this state. Once duration is extended, the long-term memory effect of the RNN will not be achieved. This is because the values to be returned will decrease in an exponential manner, resulting in a slow update of the network weights. To solve this problem, LSTM is introduced. LSTM preserves essential features through the "gate" structure to ensure they will not be lost in the long-time series propagation. This paper adopts the standard architecture of LSTM to process the high-order feature sequence further and realize the prediction. Set the time step to 2 and the hidden state to 30. LSTM is followed by a BN layer to optimize the output of the LSTM. And the output result is fed to the output layer. The activation function of the output layer is linear. Linear is a Linear transformation of the input $X_{d\times i}$:

$$Y_{d\times o} = X_{d\times i}W_{i\times o} + b \tag{5}$$

Where W is the parameter to be learned by the model, the dimension of W is $W_{i\times o}$,b is the vector bias in o dimensions, d is the dimension of the input vector, the size of o is the number of output neurons, and i is the number of input neurons. $(XW)_{d\times o}$ Afterwards, the vector b will be added to the corresponding position d vectors. Finally, according to the model flow in Figure 1, the process of learning and optimization of the forecast model is summarized in Algorithm 1.

```
Algorithm 1 Optimize the prediction model of this paper within the sample
Input：Input samples $x \in X^{n\times 9\times 30}$，The array pairs are num，num-rev, True value as target y
Output: predicted labels for the input $\tilde{y}$
1：denotes that the prediction model with all parameters as W；
2：Initialization parameters W；
3：for epoch in range (epoch_num ):
4:     $h = Z - Score(x)$；
5:     $s = $ Customized network layer $(h, num, num - rev)$ ；
6:     a=LSTM (s)；
7:     $\tilde{y} = f_{prediction}(a)$ ；
8:     Calculate the loss of batch samples；
9:     $L = LossFunction(y, \tilde{y})$;
10:    Use gradient descent $(L|W)$, Update hidden parameters W；
11：End;
```

4. Experiments

4.1. Extraction of stock data features

4.1.1 Data

This paper collects daily quotation data of all stocks on the China A-share market as of 2020, excluding ST stocks and stocks with up and down stops and suspensions. Stock data quotes include nine stock characteristics: closing price, low price, high price, opening price, volume, turnover rate, volume-weighted average price, free-float turnover rate and yield. Each stock is set into a 9*30 two-dimensional data structure, and the stock characteristics can be arranged in any order. In this study, the datasets are partitioned 3:1 in chronological order, with the training set in front and the test set in the back.

4.4.2 Model parameters

The average squared error is used in the whole model training. With this parameter being the average of the summary of the squared errors of the predicted data and the corresponding points of the original data, the calculation formula is:

$$MSE = \frac{1}{n}\sum_{i=1}^{m} W_i \left(y_i - \widehat{y_t}\right)^2 \tag{6}$$

y_i is the actual data, $\widehat{y_i}$is the fitted Data, and $w_i > 0$, the size of n is the sample size. In this study, the exponentially decaying learning rate is used to optimize the learning rate of the model, and the initial learning rate is chosen to be 0.05.In the design of the output layer, initialization of the weights with the normal distribution, where the random numbers are taken from the normal distribution with a mean of 0 and a standard deviation of 0.01, and the bias is initialized to 0. Each sample's standard deviation and mean are calculated with the z-score normalization function, so that the raw inventory data from all baselines are normalized.Batch Normalization (BN layer) was added after each custom network layer. Before BN layer normalization, the custom network layer extracted features with values ranging from(-12.5,2.5), a wide range. After BN normalization, thecontent of features is very close to the field (-2,3), which is very useful for model training. The histograms of the feature distributions are shown in Figures 4and 5.

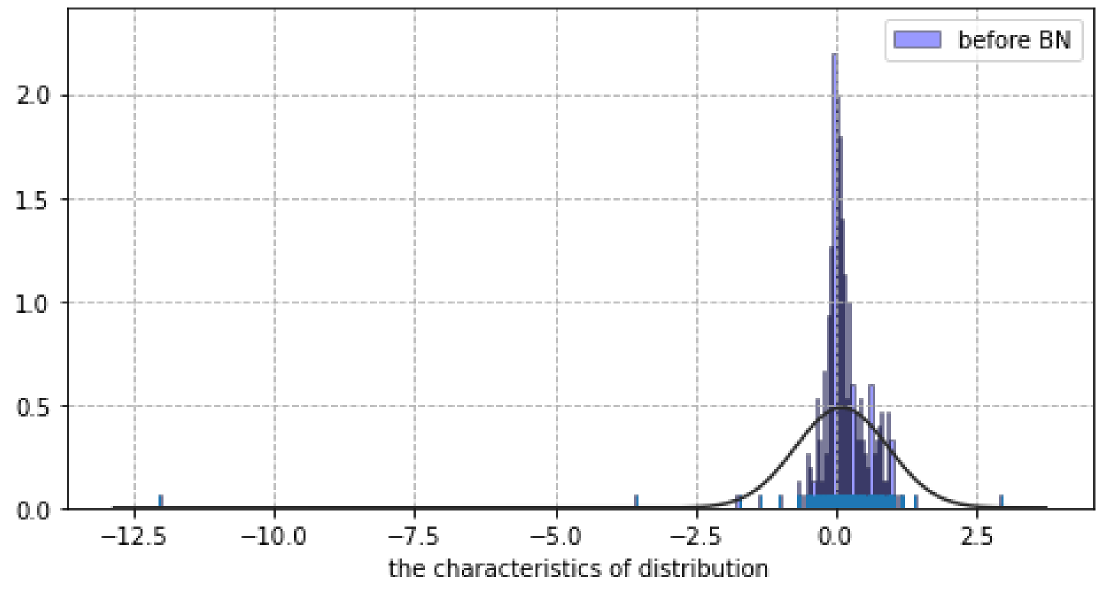

Figure 4: Distribution of features before BN

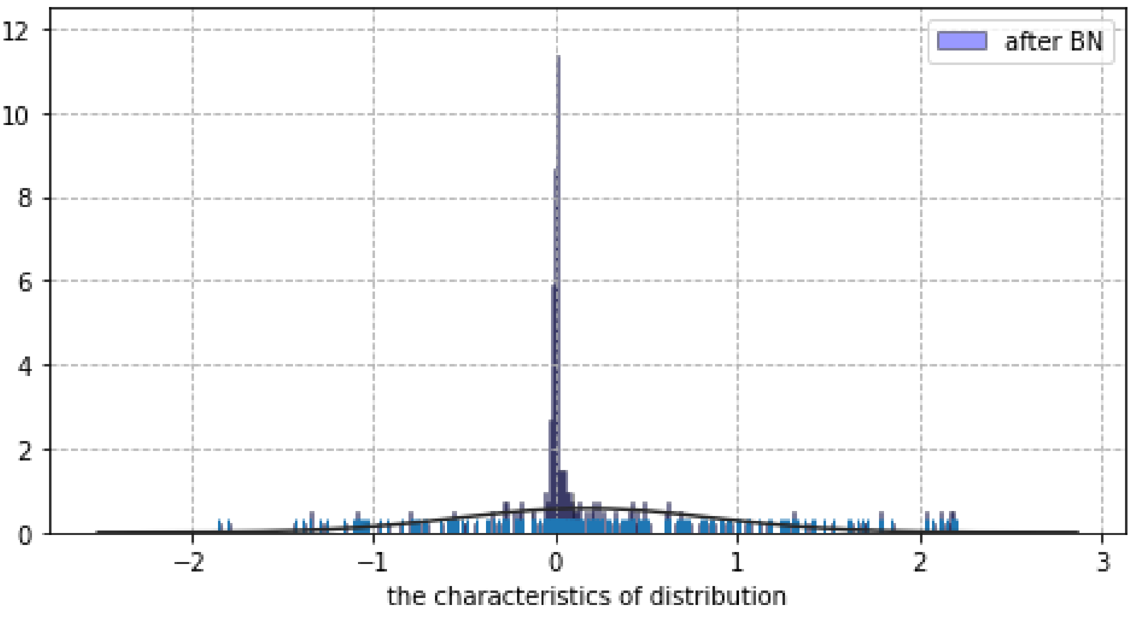

Figure 5: Distribution of features after BN

4.2. Experimental results and discussion

4.2.1 Experimental results

When testing the model's performance, RMSE, MAPE, SMAPE and correlation coefficient are used as performance measures. LSTM model and one-dimensional convolution optimized LSTM model (Conv1D+LSTM) are used to compare with the model constructed in this paper (Custom+LSTM). The data of 1436 stocks were input into the model for training, and the opening price of the stock is used to predict the next day's upward or downward trend. In this study, stochastic gradient descent (SGD) and adaptive momentum optimization (Adam) have been tested separately.The Adam optimizer gives superior results than the SGD optimizer, so the Adam optimizer is chosen. The outcomes are listed in Tables 2 and 3.

Table 2: Performance metric results using SGD optimizer

Model	RMSE (SGD)	MAPE (SGD)	SMAPE (SGD)	Correlation coefficient (SGD)
LSTM	0.8258	1.90%	1.66%	87.06%
Conv1D+ LSTM	0.5719	2.80%	1.06%	77.27%
Custom+ LSTM	0.0329	29.11%	1.99%	95.03%

Table 3: Performance metric results using the Adam optimizer

Model	RMSE (Adam)	MAPE (Adam)	SMAPE (Adam)	Correlation coefficient (Adam)
LSTM	0.8278	4.17%	1.81%	87.19%
Conv1D+ LSTM	0.5704	2.23%	1.05%	87.03%
Custom+ LSTM	0.0213	0.96%	1.87%	94.64%

The LSTM model and the optimized LSTM model by one-dimensional convolution have a linear activation function. The model's loss function is all mean square error. Settings of hyperparameters in the model are identical to the values in the above experiments. When training the LSTM model, the training data was large, which resulted in an unprocessable model, so not all stocks were actually fully input when training the LSTM model, and even without processing all the data, the model was trained very slowly and with a prediction accuracy of 40%, which is still a long way from the ideal model. The accuracy of the 1D convolution-optimized LSTM model to predict the trend is significantly improved by 10% relative to the LSTM model, and it achieves partial trend judgment for the stock's opening price, however, the results were not satisfactory. Compared with the two former models, the prediction accuracy of the model constructed in this paper reaches 62%, while for the stock's opening price to achieve a part of the trend fit entirely, and in some nodes can also make up and down trend prediction.

4.2.2 Discussion of experimental results

In this study, performance metrics of three results are investigated and classified. The LSTM model with one-dimensional convolutional optimization has been shown to have superior prediction performance than the LSTM model. In contrast, the MAPE of the model constructed in this study is reduced by 3.2% and 1.27%, the RMSE is reduced by 0.8065 and 0.5491, and 7.45% and 7.61% improve the correlation coefficient, respectively, compared with the LSTM neural network and the one-dimensional convolution-optimized LSTM model, this indicates that the forecasting model used in this study has good forecasting accuracy and generalization ability. The performance measures of the three models are shown in Figures 6, 7, and 8 respectively.

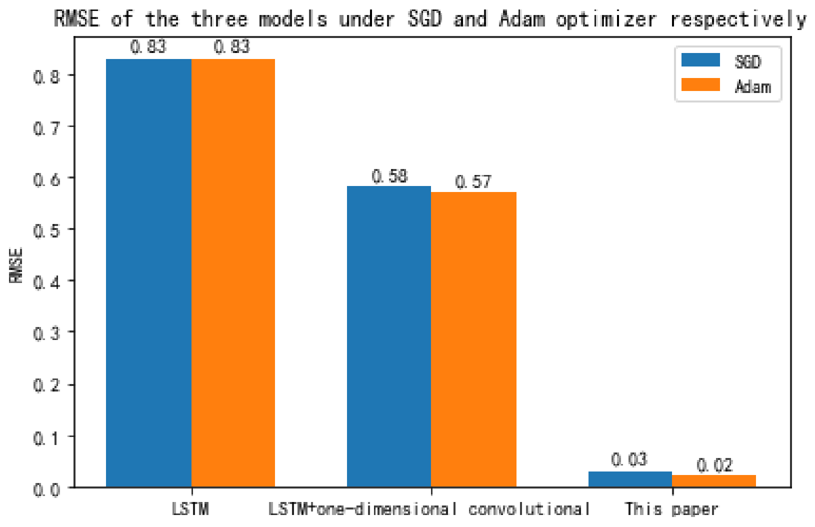

Figure 6: RMSE for each of the three models

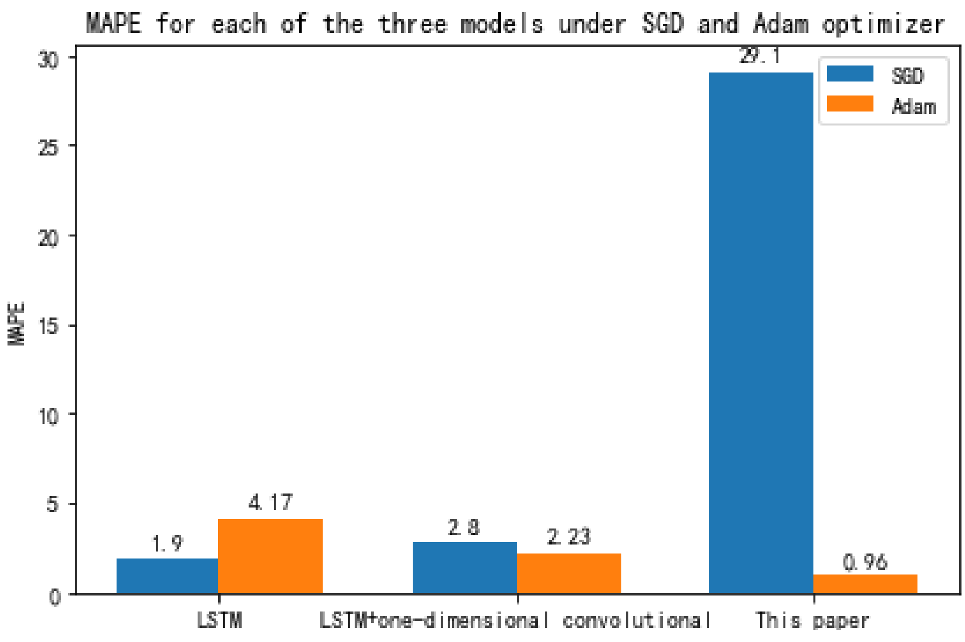

Figure 7: MAPE for each of the three models

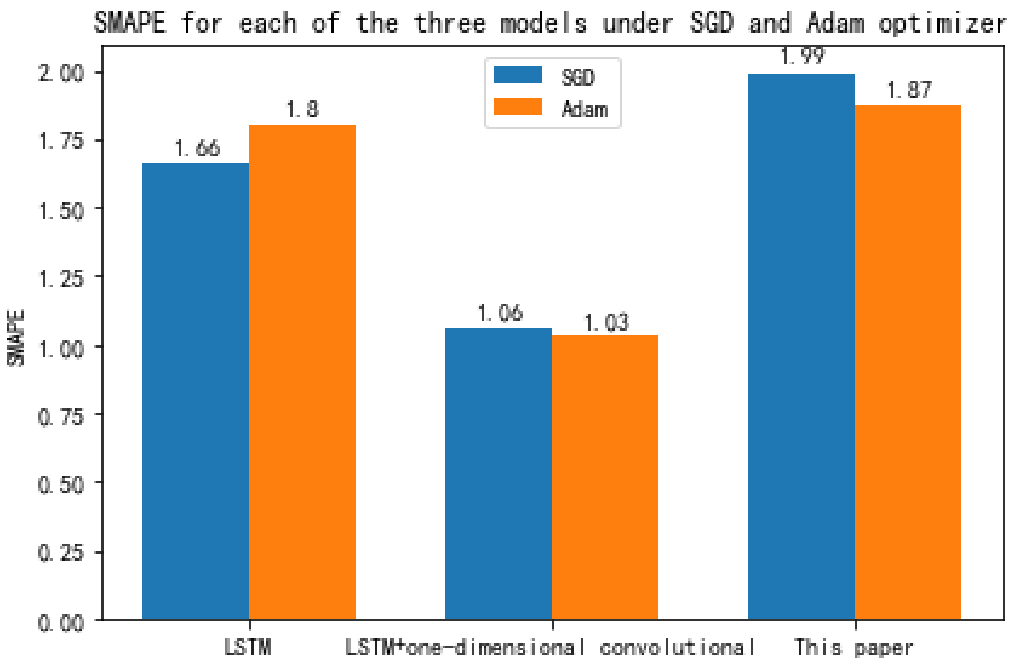

Figure 8: SMAPE for each of the three models

5. Conclusion

In this research, it combines different characteristics of stock data to forecast stock movements. First, a feature extraction layer is built and optimized for feature extraction from stock data. Secondly, long-term dependence in the time series is captured by LSTM. Finally, experiments were conducted. RMSE, MAPE, SMAPE and correlation coefficient were used as performance metrics to ensure that the proposed model is meaningful. The results show that this model outperforms the other two models in terms of data processing speed and size. Thus, it is effective to construct a custom network to extract features and the model constructed in this work is important in the forecasting of stock trends.

Although, the model built in this study has some improvements over the LSTM model and the one-dimensional convolutional optimization LSTM model, it cannot predict the stock trend completely and accurately.Therefore, finding other methods to improve the model's performance is necessary. Next, we will add more stock features to improve stock information. Continue to study and add a custom network that can better summarize the stock features to extract more comprehensive features; The gated cyclic unit GRU[18][19] was used to replace the LSTM.

References

[1] Zhang L, Aggarwal C, Qi G J. Stock Price Prediction via Discovering Multi-Frequency Trading Patterns[C]// Acm Sigkdd International Conference on Knowledge Discovery & Data Mining. ACM, 2017:2141-2149.

[2] Yan X, Guosheng Z. Application of kalman filter in the prediction of stock price[C]//5th International Symposium on Knowledge Acquisition and Modeling (KAM 2015). Atlantis Press, 2015: 197-198.

[3] Peter G, Zhang. Time series forecasting using a hybrid ARIMA and neural network model [J]. Neurocomputing, 2003.

[4] Parzen E. ARARMA models for time series analysis and forecasting [J]. Journal of Forecasting, 2010, 1.

[5] Keim D B, Stambaugh R F. Predicting returns in the stock and bond markets [J]. Journal of financial Economics, 1986, 17(2): 357-390.

[6] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[7] Anandarajan P M. Classifying inventory using an artificial neural network approach [J]. Computers & Industrial Engineering, 2002.

[8] Nelson D, Pereira A, Oliveira R. Stock market's price movement prediction with LSTM neural networks[C]// 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, 2017.

[9] Chen K, Zhou Y, Dai F. A LSTM-based method for stock returns prediction: A case study of China stock market[C]// IEEE International Conference on Big Data. IEEE, 2015:2823-2824.

[10] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition [J]. Computer Science, 2014.

[11] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.

[12] Hoseinzade E, Haratizadeh S. CNNpred: CNN-based stock market prediction using a diverse set of variables [J]. Expert Systems with Applications, 2019, 129(SEP.):273-285.

[13] Di Persio L, Honchar O. Artificial neural networks architectures for stock price prediction: Comparisons and applications [J]. International journal of circuits, systems and signal processing, 2016, 10(2016): 403-413.

[14] Hu Z, Liu W, Bian J, et al. Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction: ACM, 10.1145/3159652.3159690 [P]. 2018.

[15] Xu Y, Cohen S B. Stock movement prediction from tweets and historical prices[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018: 1970-1979.

[16] Bordino I, Kourtellis N, Laptev N, et al. Stock trade volume prediction with yahoo finance user browsing behavior[C]//2014 IEEE 30th International Conference on Data Engineering. IEEE, 2014: 1168-1173.

[17] Vantuch T, Zelinka I. Evolutionary based ARIMA models for stock price forecasting[C]//ISCS 2014: Interdisciplinary Symposium on Complex Systems. Springer, Cham, 2015: 239-247.

[18] Zhang X, Shen F, Zhao J, et al. Time Series Forecasting Using GRU Neural Network with Multi-lag After Decomposition[C]// International Conference on Neural Information Processing. Springer, Cham, 2017.

[19] Selvin S, Vinayakumar R, Gopalakrishnan E A , et al. Stock price prediction using LSTM, RNN and CNN-sliding window model[C]// International Conference on Advances in Computing. IEEE, 2017.

[20] Cheng C H, Chen T L, Wei L Y. A hybrid model based on rough sets theory and genetic algorithms for stock price forecasting [J]. Information Sciences, 2010, 180(9):1610-1629.

[21] Kim K J, Han I. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index [J]. Expert Systems with Applications, 2000, 19(2):125-132.