Stock Price Prediction Based on Decision Trees, CNN and LSTM

Ruotong Li [1, †] Moying Ma [2, *, †] Nan Tang [3, †]

531887691@qq.com[1], *15220182202589@stu.xmu.edu.cn[2], tangn27@mail2.sysu.edu.cn[3]

Department of electronic commerce, South China University of Technology, Guangzhou, China[1]
Department of Economy, Xiamen University, Fujian, China[2]
School of Marine Sciences, Sun Yat-sen University, Guangzhou, China[3]
† These authors contributed equally.

Abstract—Stock price forecasting plays an important role in quantitative transaction and financial analysis. In this paper, three well-known machine learning approaches, decision tree, LSTM, and CNN, are implemented in order to realize stock price prediction. After introducing each model's background, principle, and method, the historical closing price of China Merchants Bank stock is used as the training set data. After the establishment of the three models, the comparison between the predicted value and the real value on the test set is demonstrated. According to the results, the prediction feasibility is verified to a certain extent. Furthermore, it is concluded that the forecast effect in the short term is better than that in the long term. These results shed light on the effectiveness limitations of machine learning used in quantitative trading.

Keywords-Stock price; forecasting; decision trees; CNN; LSTM

1. INTRODUCTION

The stock market plays an important role in China's stock market. To a certain extent, the stock price trend can reflect the future development trend of the national economy. Meanwhile, with the development of society, stock investment plays an increasingly important role in people's life. Therefore, the fluctuation of stock price directly affects the development of stock market and social economy. Stock price prediction can help investors to make investments and government agencies to guide and supervise the market. Therefore, the stock price forecast has very important theoretical significance and practical value. However, the path toward forecasting the stock price is a challenging subject existing in the financial circle. Plenty of scholars use a variety of machine learning models [1] to predict the stock price. Nevertheless, as the financial market is a complex system with multiple agents, the stock price trend is affected by many factors. Therefore, based on the multi-factor stock price forecast method, this paper provides a more powerful technical support for the stock price forecast.

ICEMME 2022, November 18-20, Nanjing, People's Republic of China

DOI 10.4108/eai.18-11-2022.2327160

Machine learning is a data analysis method that does not depend on rule design. Through repeated iterative training, this method can discover the rules hidden in the sample data and mine the intrinsic value of the data [2]. The decision tree [3, 4] is a classical algorithm in machine learning, which possesses advantages such as fast classification speed, high readability [5], and excellent performance in stock price prediction. Specifically, Zhao applied the decision tree classification algorithm to the analysis of stock financial data, selected representative financial indicators, and tested the sample data. Investors can use the test results to analyze listed companies' operating conditions and profitability [6]. This paper starts from the direction of stock technical analysis (the strategy of predicting price trend by studying information of the past financial market [7]) and proposes the research on stock price trend based on the optimal decision tree model. Furthermore, state-of-art machine learning models (e.g., LSTM and CNN) are also applied for better comparison. This article uses the decision tree model, LSTM model and CNN model to predict the trend of the stock price of a single stock (China Merchants Bank).

The rest part of the paper is organized as follows. Sec. 2 introduces the data origination and training method of the decision tree, LSTM, and CNN. Sec. 3 presents the training results and corresponding discussions. Finally, a brief summary is given in Sec. 4.

Figure 1. Closing price chart of China Merchants Bank stock (2011-07-26 to 2021-07-26)

2. DATA & METHOD

2.1 Data

We select the historical data of China Merchants Bank from NetEase Finance for the entire ten years daily data from July 26, 2011 to July 26, 2021, as the sample set (illustrated in Fig. 1), which is downloaded directly on the official website of Netease Finance. After data processing, the shape of the data is [2432 rows x 13 columns], where 13 columns correspond to ‘date’, ‘open’, ‘close’, ‘high’, ‘low’, ‘previous close’, ‘price_change’, ‘p_change’, ‘turnover’, ‘trading volume’, ‘transaction amount’, ‘total market value’, ‘circulation market value’.

2.2 Decision tree

As a commonly used machine learning algorithm, decision tree has two functions of classification and regression, and even performs multi-output tasks. As a powerful machine learning algorithm, it can process complex and high-dimensional data.

Decision tree method is a kind of time series forecasting model, which is an inductive learning algorithm based on examples. Besides, it classifies a set of unordered and unregulated cases, and represents them as a "tree", using a top-down recursive approach. Moreover, attribution selection is performed on the internal nodes of the decision tree, and the decision tree is pruned. Decision tree algorithm description is simple, modeling speed is fast, and prediction results are easy to interpret. This article's decision tree classification method is the C5.0 algorithm, which is a step-by-step development and improvement from the originator of the decision tree algorithm ID3 algorithm to C4.5 algorithm. After improvement, its comprehensive performance is greatly improved.

Generally, the growth and division of the decision tree mainly start from two aspects: first, how to choose the current best split variable from the many input variables; second, how to find the best split point from the many values of the current split variable. As for determination of split attributes, C5.0 decision tree uses "gain ratio" as the split attribute of the current node to measure the breadth and uniformity of the data. Suppose S is a training sample set, which consists of s samples, including m different classes x_i (i=1, 2, ..., m), the number of samples of each class x_i is v_i, D is an attribute of the training sample set S, which has k different values. According to these values, S can be divided into k different subsets. S_i represents the i-th subset (i=1, 2, ..., k), and s_i represents the number of samples in the subset S_i, then the information gain $Gain(S, D)$ can be expressed as:

$$Gain(S, D) = I(s_1, s_2, \cdots, s_m) - E(S, D) \tag{1}$$

where

$$I(s_1, s_2, \cdots, s_m) = -\sum_{i=1}^{m} p(x_i) \log_2(x_i) \tag{2}$$

which is the entropy of the sample set, $p(x_i)$ represents the probability of each category and $\sum_{i=1}^{m} p(x_i) = 1$, $E(S, D)$ represents the weighted sum of entropy of k subsets divided by attribute D. This split information item can be expressed as:

$$Split\_Info(S, D) = -\sum_{i=1}^{k} \{(|S_i|/s) \log_2(|S_i|/s)\} \tag{3}$$

This split information item is called the entropy of the data set S on the attribute D. The more uniform the value distribution of the sample on the attribute D, the larger the value of this split information item. The gain ratio can be expressed as:

$$GainRatio(S,D) = \frac{Gain(S,D)}{Split\_Info(S,D)} \quad (4)$$

Obviously, the larger the breadth of the attribute D split data set, the stronger the uniformity, the larger the split information item will be, and the smaller the gain ratio will be.

The next step is to determine the best split point. After determining that the best splitting attribute is a discrete variable with K categories, the sample set is divided into k groups, i.e., k branches of the decision tree. When the best splitting attribute is a continuous variable, according to the minimum group limit obtained by MDLP binning, the samples smaller than the threshold are divided into one group while the samples larger than the threshold are divided into another group, i.e., the formed decision tree contains two branches.

The above process is repeated until the grouping is continued, and the grouping is stopped when it is meaningless. Combined with the subsequent pruning work, the best decision tree classifier can be obtained. When using the decision tree model to predict stock prices, it should meet the following conditions from mathematical statistics.

Primarily, there ought to be a linear relationship between the response variable y and the predictor variable x. Therefore, it is more reasonable to explain the data through a line (when x is only one-dimensional) or a surface (when x is high-dimensional) as shown in Fig. 2.

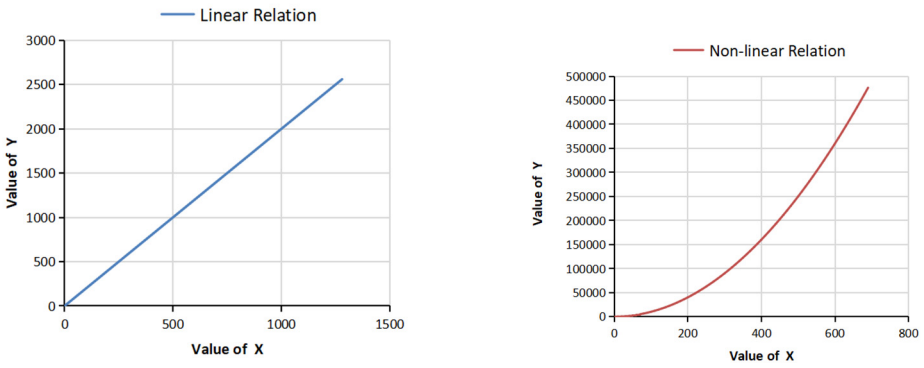

Figure 2. Linear & Non-liear relationship.

The second requirement is the independence of the error term, i.e., different observations should be independent of each other. However, the actual situation is that today's stock price may be related to the previous day's stock price. If independence is violated at this time, the following situation will result:

- the variance of the independent error is smaller than the variance of the dependent error;
- after the variance is underestimated as the denominator, the t statistic will become larger. Although the estimation of parameters is unbiased, it is inefficient and consistent;

- the third is homoscedasticity, i.e., the variance of each error term is the same.

Violating this property will cause the following effects:

- the estimated value of variance is biased, and the estimated value of variance derived from the OLS assumption will not change, but in fact it will change;
- parameter estimation is unbiased and consistent, but inefficient;
- the established model is unstable and the original data is lost;

Besides, normality is also important. This theory actually does not affect the stability and accuracy of the model even if it violates the huge amount of data. Due to the confirmation of the law of large numbers: in the process of increasing sample size, the accuracy of the calculation of the p-value will gradually increase.

Since we need the date of the stock as the basic index, which is arranged in ascending order of time, we convert the total string of the key value type of each data about the date into a date, and then check whether there are missing data NaNs. In fact, only the stocks are not traded on weekends, and the data of China Merchants Bank stocks on working days are relatively comprehensive. Then we use the third-party library matplotlib and tools in python to draw the K-line graph of the data. Among them is stock data for the ten years from 2011-07-26 to 2021-07-26, including opening and closing prices and high and low prices.

Next is the linear regression part, mainly using the sklearn.linear_model in the third-party library LinearRegression, and define a label and move the closing price forward by num days; then delete price_change and p_change to get the new data Data; set the value of Data to X and the label to Y.

The next work is to divide the sample set into a training set and a test set, the basic ratio of which is about 3:1. And use the absolute correlation coefficient to determine the fitting effect. After the regression, we define the predicted value as forecast and add it to the original pandas.Dataframe. Finally, the true value close and forecast value forecast are drawn in a line graph with Time as the horizontal axis and Price as the vertical axis. From the absolute correlation coefficient and the line graph, we can compare the prediction effect.

2.3 Long Short-Term Memory Model M

Recurrent neural network (RNN) is a branch of artificial neural network, in which the connections between nodes form a directed graph along time series, allowing it to display real-time dynamic behavior. Unlike feedforward neural networks, RNN gives the network a “memory” of the past content and processes the subsequent input sequences based on this memory. This advantage enables it to have good application scenarios in text recognition, speech recognition and image processing [8].

Although the traditional neural network model overcomes difficulties with the nonlinear function, with good ability to simulate, its simulation lacks dynamism. Its theoretical assumption is that the inputs and outputs are independent of each other. However, this assumption is not established in the real world. Fortunately, the recurrent neural network doesn’t rely on this assumption. It performs the same task for each element in the sequence, and the calculation of the output layer depends not only on the input layer but also on the previous

calculation results. Each time it performs an operation, it automatically remembers the previous result, and in theory, it can capture all the previous information [8].

Nevertheless, RNN structures have certain defects. The entire RNN structure shares the same set of neuron formula parameters. Under the condition these parameters are unchanged, the gradient is continuously multiplied in the process of back propagation, and the value will become much larger or smaller, so the training model with RNN cannot achieve the expected effect.

Compared with RNN, LSTM has a different internal structure and different calculation formula of neuron. The neurons of LSTM are added with input gate, forgetting gate, output gate and internal memory unit. Based on such gates, this model can control the inflow and outflow of some information and prevent data explosion, which is better than RNN. Each of its doors has its own neuron formula parameters.

If there is no useful information in the input sequence, the past useful information can be saved by adjusting the forgetting and input gate values. If there is useful information in the input sequence, the forgetting gate and input gate values are adjusted accordingly. In this case, the LSTM model forgets the past memories and records important memories.

Designed by forgetting gate, input gate, output gate and internal memory unit to control the output of LSTM, the whole LSTM network can better grasp the relationship between sequence information.

2.4 Convolutional Neural Network

Convolutional neural network, CNN model, is one of the representative algorithms of deep learning. Its basic structure consists of input layer, convolution layer, pooling layer, full connection layer and output layer. The reason is that each neuron of the output feature map in the convolution layer is locally connected with its input map, and weighted sum with the local input plus the bias vector, which is equivalent to the convolution process, i.e., this model is named CNN [9].

In the convolution layer, each feature map is composed of multiple neurons, and each neuron is connected with a local region of the feature map from the upper layer through convolution kernels. The convolution kernel is a weight matrix, which extracts different input features through convolution operation. The pooling layer follows the convolution layer, which aims to obtain the features with spatial invariance by reducing the resolution of the feature surface, and plays the role of secondary feature extraction. After the convolution and pooling layers, more than one full connection layer may integrate local information in the convolution layer or pooling layer. The output value of the last full connection layer is passed to the output layer [9].

Taking the shape of stock price wave as the feature, different convolution kernels are used to extract different features. Besides, the data of the first 20 days are used to predict the data of the next day. After the adjustment, the total number of data used is 2429, of which the first 70% is used as a training set and the last 30% is used as a test set. The adopted index is the adjusted closing price which later is adjusted to [0, 1].

After understanding the principle, this paper uses Keras to build the model. Keras is written in pure python and advocates minimalism, which makes users use less code to realize various

ideas. Therefore, Keras is a highly modular neural network library. There are two ways to construct CNN in Keras. The first method is the sequential model, which is a linear stack of multiple network layers. However, if there is more than one output in the model, the second method should be chosen, that is, the functional model [10]. In this paper, because only the basic CNN model is needed and the output is a simple result, the sequential model is used for convenient operation.

The following is a brief introduction to the model construction. First, there is a convolution layer, in which the size of convolution kernel is 4 and the number is 50. In the meantime, the model uses zero filling and the activation function is "ReLU". Then, there is a pooling layer. The pooling method takes the larger one for every two data. Subsequently, a flatten layer is used to make the input one-dimensional, which is often used in the transition from convolution to full connection layer. Finally, there is a full connection layer, and the optimizer is SGD. Moreover, this paper will use Dropout to prevent overfitting.

3. RESULTS & DISCUSSIONS

3.1 Decision trees

We use the data from 2011-07-26 to 2021-07-26 as the sample set, set the training set and the test set at a ratio of 2 to 1, and then perform data training to predict the closing price of stocks in the last 30 days of the sample set. It can be seen from Fig. 3 that in this month, the forecast values and the real closing price close can be fitted to a certain extent in the numerical value and trend.

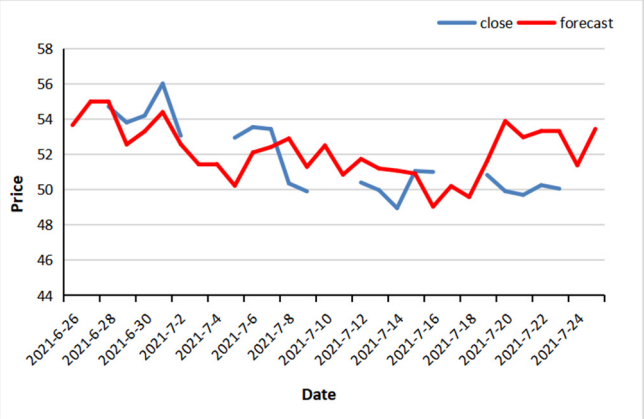

Figure 3. 2021-06-26 to 2021-07-26 closing price and forecast.

The model is metric by R^2, i.e., the absolute correlation coefficient, to evaluate the fitting efficiency of the linear model. Figure 4 is the result of training under the condition of R2 = 0.83, i.e., it can be judged to a large extent that the prediction sample is 15 days outside Changes in stock prices.

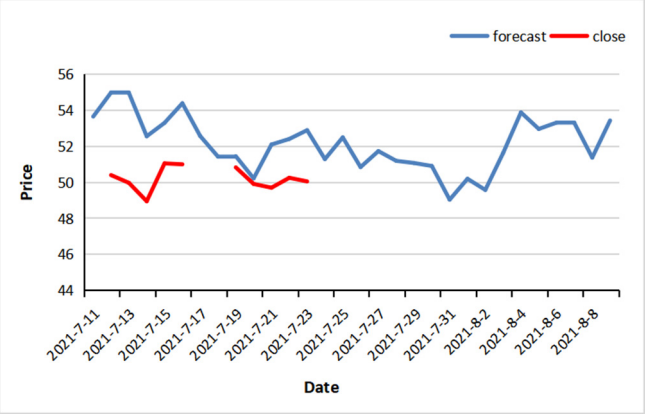

Figure 4. Out-of-sample stock price forecast.

After we finally obtain the prediction results, one can get each variable as the linear relationship between the independent variable X and the dependent variable stock price Y. Each independent variable and its coefficient are represented in Table I (retain two significant digits after the decimal point).

TABLE I. TABLE TYPE STYLES

Coefficient	Value
open	5.77
high	3.41
close	0.17
low	-10.01
volume	10.57
turnover	-3.61
trading volume	4.87
transaction amount	-1.62
total market value	35.32
Circulation market value	-32.91

Subsequently, one can derive the final linear equation:

$$Forecast = 5.77 \times Open + 3.41 \times High + 0.17 \times Close - 10.01 \times Low + 10.57 \times Volume - 3.61 \times Turnover + 4.87 \times Trading\_volume - 1.62 \times Transaction\_volume + 35.32 \times Total\_market\_value - 32.91 \times Circulation\_market\_value \quad (5)$$

3.2 LSTM

After calculating the prediction error, the training error score is 1.627, and the testing error score is 4.123. It can be seen that the error is small, and the comparison result is ideal. After sorting out the drawing data, the results are obtained as illustrated in Fig. 5

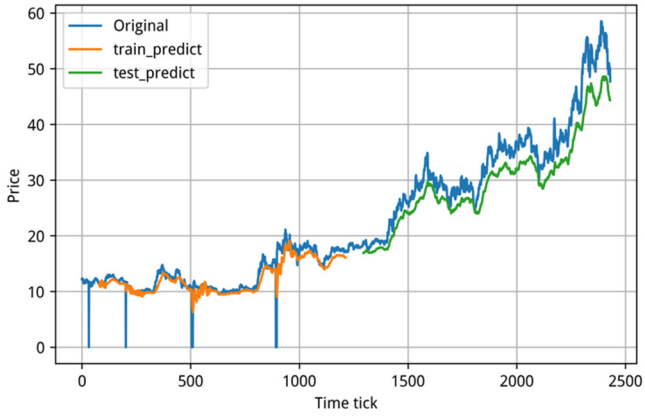

Figure 5. Comparison of predicted and true values

According to the results, the test price is close to the forecast price. Based on LSTM neural network model and Adam optimization algorithm, the historical data of stock price are designed and modeled. On this basis, the model can achieve a good prediction effect, which verifies the model's validity. Thus, it has a certain guiding significance for investment.

3.3 CNN

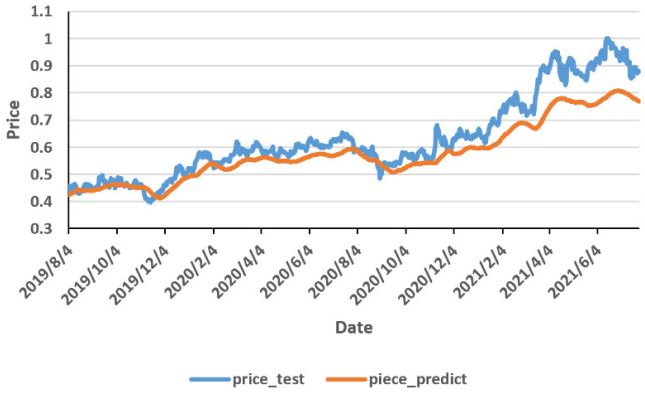

Figure 6. Overall Effect of Stock Price Forecast.

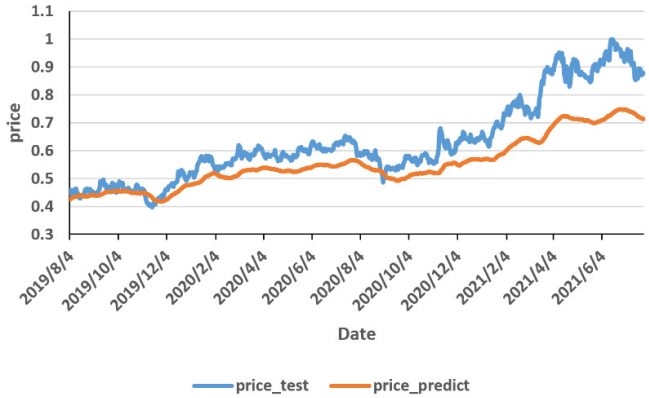

Figure 7. Overall Effect of Stock Price Forecast after Increasing Kernels to 60.

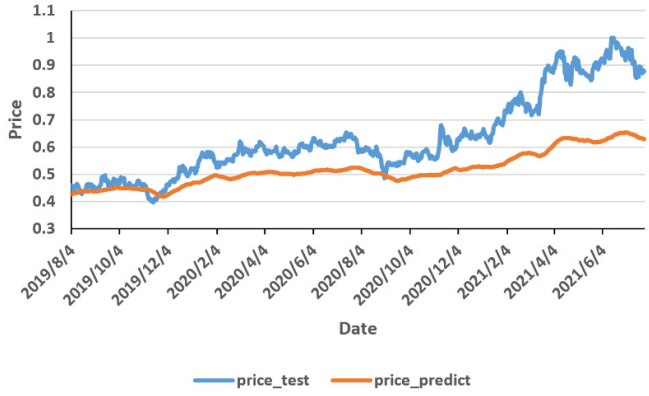

Figure 8. Overall Effect of Stock Price Forecast after Increasing Kernels to 70.

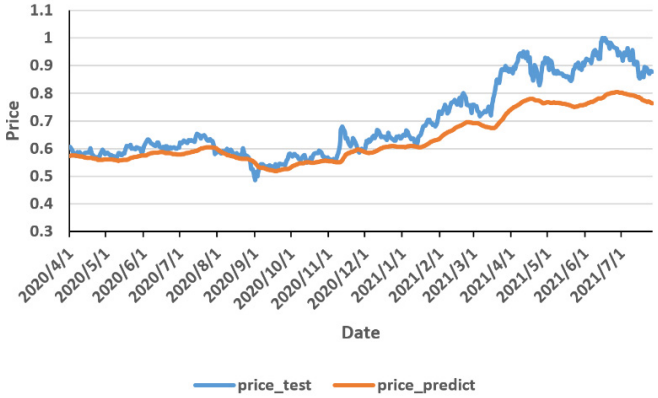

Figure 9. Overall Effect of Stock Price Forecast after Increasing Training Set to 80%

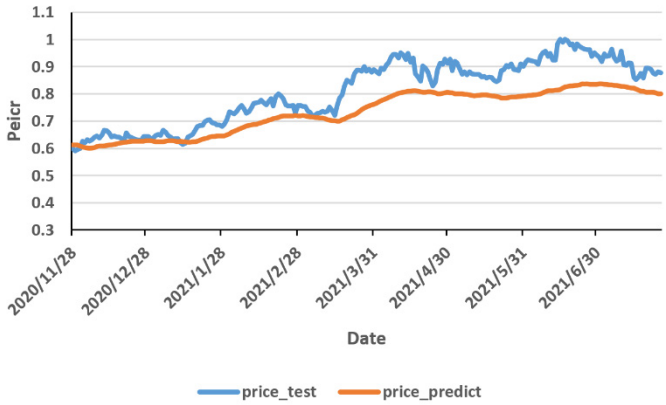

Figure 10. Overall Effect of Stock Price Forecast after Increasing Training Set to 90%

As shown in Fig. 6, when the number of convolution kernels is 50, and the size of the training set is 70%, the loss value of the test set is about 0.0098. As depicted in Fig. 7, when the number of convolution kernels is 60, and the size of the training set does not change, the loss value is about 0.0084. As displayed in Fig. 8, when the number of convolution kernels is 70 and the size of the training set does not change, the loss value is about 0.0078.

As seen from Fig. 9, when the number of convolution kernels is 50 and the size of the training set is 80%, the loss value of the test set is about 0.0036. As exhibited in Fig. 10, when the number of convolution kernels is 50, and the size of the training set is 90%, the loss value is about 0.0223.

Firstly, the number of convolution kernels is one of the parameters of CNN, so this paper changes the number of kernels to carry out comparative experiments. According to Figures 6-8, when the number of kernels is increased, the loss value is reduced by 0.002, and the accuracy of the model is slightly improved. However, with the increase of kernels, the time consumed increases. Therefore, to improve the accuracy, we must also take into account the efficiency of training, and we cannot blindly increase kernels.

Secondly, on improving learning accuracy, many materials talk about the method of increasing the size of the training set, so this paper expands the training set to compare with the previous results. As illustrated in Figures 6, 9 and 10, when the expanded training set is 80%, the loss value decreases by 0.0062, but while the training set is 90%, the loss value increases by 0.0125, which is inferior to the performance of the previous small training set.

The characteristics of financial data may cause this, that is, the stock price reflects the market changing at any time, i.e., an internal law of the market may be relatively obvious only in a short cycle, which means it is not easy to find the law of recent performance from a large training set. This is different from picture recognition using CNN, whose features are fixed, i.e., the accuracy can be improved after adding samples [10].

In addition, according to behavioral finance, some information is not related to asset value but may affect stock price in the market. It makes the stock price deviate from the equilibrium value to a certain extent. Irrational changes follow such information. Therefore, introducing

more samples also brings more information, which is likely to drown out the real law. This result suggests that when predicting the stock price, the prediction result obtained by using a training set in a short period may be better [10].

4. CONCLUSION

In summary, this paper uses decision trees, LSTM models and convolutional neural network models to quantify trading as the main research purpose. The three models are used to perform data acquisition, modeling and parameter adjustment, and stock price prediction for a single stock (China Merchants Bank). When selecting data with a larger time span, it has a strong ability to predict the trend of a single stock. Therefore, stock prices predict that shortening the period will improve the accuracy of the model. Additionally, it will also better derive the trend of the closing price of the stock. These results reveal the effectiveness and limitations of machine learning in the field of quantitative investment.

REFERENCES

[1]Nguyen D H, Le M T. A two-stage architecture for stock price forecasting by combining SOM and fuzzy-SVM[J]. (IJCSIS) International Journal of Computer Science and Information Security, 2014,12(8): 1-6.

[2]Bai Mingrui. Research on stock price prediction based on optimized GAM model[D]. Northwest Normal University, 2020.

[3]Giovannis S, John E. Ensemble Methods in Data Mining: Improving Accuracy Through Combining Prediction[M]. Morgan and Claypool, 2010: 28.

[4]Aurelian Geron. Hands on Machine Learning with Scikit-Learn & Tensorfolw[M]. Beijing: Machinery Industry Press, 2020:185-200

[5]Paulius D, Gintautas G. Selection of Support Vector Machines based classifiers for credit risk domain[J]. Journal of Expert System with Applications, 2015, 42: 3194-3204.

[6]Zhao Yongjin. Research on Stock Analysis and Forecast Based on Data Mining[D]. Zhengzhou University, 2005.

[7]John Murphy. Financial market technical analysis[M]. Seismological Publishing House, 2010.

[8]Xu Tiantian. Research on Prediction of Stock Price Rise and Fall Based on LSTM Neural Network Model[D]. Shanghai Normal University, 2019.

[9]Zhou Feiyan, Jin Linpeng, Dong Jun. Review of convolutional neural networks [J]. Journal of Computer Science, 2017,40(06):1229-1251.

[10] Chen Xiangyi. Prediction of CSI 300 index based on convolutional neural network [D]. Beijing University of Posts and Telecommunications, 2018.