Research on Stock Market Index Prediction Based on LSTM Model

Changquan Gu*

School of Intelligent Systems Science and Engineering

Jinan University 519070, Zhuhai, China

*Corresponding author: guchangquan0406@163.com

Abstract. In this paper, we propose a method based on a long short-term memory network (LSTM) to predict the rise and fall of a stock market index, and we select the data of CSI 300 from 2016 to 2020 as the research object of this paper. In this paper, the prediction problem of the index is transformed into a multi-classification problem by doing a multi-valued quantitative classification of the up and down ranges. As for the input of the model, we take the basic trading information of the stock and a variety of derived technical indicators as the feature for the model's training, and finally, we make the multi-classification prediction of the rise and fall of the stock. The experimental results show that the model achieves a high prediction accuracy of 78% in the case of multiple classifications. In addition, several comparison models are built in this paper to study the effects of different feature engineering methods on prediction accuracy, so there are some innovations in feature construction. It is concluded that filtering the features before dimensionality reduction can achieve better performance results. Finally, we compare the method proposed in this paper with other traditional machine learning models and conclude that the LSTM model in this paper yields better results while using the same training and test sets.

Keywords: Stock Forecasting; LSTM; Multi-Classification; Feature Engineering.

1. Introduction

Stock trading is an important way for people to make profitable investments, and stock price volatility focuses on investors' attention and research. With a total market capitalization of more than 106 trillion dollars at the end of March 2021, fluctuations of global equity markets have a significant impact on the global economy. Since the inception of the stock market, numerous researchers have analyzed the movements of stock prices to find patterns in their movements and forecast their trends. Many widely recognized studies have proved that financial markets can be predicted to some extent [1-2]. However, its nonlinear instability and complexity make it challenging to study the prediction of stock prices. In the early days, experts used fundamental and technical analysis to predict stock prices, the latter of which was based on the Dow Theory [3].

The methods used in predicting stock returns can be divided into linear and nonlinear models. The traditional nonlinear model, exemplified by the ARIMA model, has been used by many scholars for stock price forecasting [4-5-6]. However, because stock data contain a lot of noise and uncertainty factors, the own limitations of linear models keep highlighting as the prediction period becomes longer. White [7] was the first to propose applying artificial neural network models to stock market forecasting in 1988, but the prediction results were not very satisfactory. Pan [8] and other scholars started to focus on applying multilayer perceptron MLP to stock forecasting. Later, more and more hybrid models were applied to stock market forecasting [9-10-11]. Most of the experimental results indicate that the mixed model can further improve the accuracy.

Most artificial neural networks (ANNs) have only a shallow structure, leading to their lack of a successful multilayer training strategy. Moreover, ANNs tend to fall into local optimal solutions and lead to overfitting. Hinton and other scholars [12-13-14] have proposed the concept of deep learning, arguing that multilayer models can better approximate nonlinear functions than shallow models.

Benefiting from the improvement of big data technology and computing performance, deep learning techniques represented by convolutional neural networks (CNNs), recurrent neural networks (RNNs), etc., have made breakthroughs in text, image, and speech video recognition, and there is a

growing interest in whether deep learning can be effectively applied to financial problems. For the application of CNNs to stock prediction, most scholars could not achieve satisfactory results, despite optimizing the inputs or models [15-16-17]. Later more scholars [18-19-20] chose LSTM, which has better application to time series, for their research and achieved lower average error and higher accuracy. Kim [21] proposed a hybrid LSTM-CNN model that incorporated features learned from various characterizations of the same data to predict stock prices, and the conclusion showed that the model outperformed the single model in predicting stock prices. However, there are still some problems to be solved for the application of deep learning in stock price prediction, among which the accuracy of the prediction is essential. The current accuracy of prediction is only at the level of recommendation and assistance, which needs to be improved.

By synthesizing the above research literature, the LSTM deep neural network model applicable to time series forecasting is selected in this paper to forecast the trend of the stock market index in the future period. In order to better reflect the stock trend situation and the basic information of daily stock trading, several daily stock technical indicators that have been practiced, summarized, revised, and adjusted by market investment theory researchers for many years are introduced in the study. Based on the different processing methods of previous studies, we first apply the random forest model to all indicators of the sample to get the importance ranking, screens out the indicators that are more important to the model, and then apply the principal component analysis for dimensionality reduction to fully optimize the sample. In addition, we combined grid search, random search, and Bayesian optimization to find the hyperparameters of the model, resulting in final prediction accuracy of 78%. Our proposed model is also optimal compared to other traditional machine learning models.

2. Data Source

The data used in this paper are collected from CSI 300 on CSMAR from January 4, 2016, to December 31, 2020. CSI 300 is a financial indicator reflecting the compilation objectives and operation status of CSI 300, which Shanghai and Shenzhen Stock Exchanges jointly released on April 8, 2005. It can be used as an evaluation standard for investment performance, providing the basic conditions for indexed investment methods and derivative product innovation. CSI 300 takes December 31, 2004, as the base date and the adjusted market value of 300 constituent stocks as the base period, with the base index set at 1,000 points, and has been officially released since April 8, 2005. Compared with the Shanghai Composite Index, commonly used in other studies and includes all stocks listed on the SSE, CSI 300 is more representative of the development of quality enterprises in China and is a better target for research.

The data set is generated by collecting data samples once a day at the stock market's close, which contains the basic indicators of the opening index, closing index, high index, low index, volume, and turnover rate, as shown in Table 1.

Table 1. Sample of raw data

Code	Date	Open	Close	High	Low	Volume	Change
000300	2016-01-04	3725.856	3469.66	3726.245	3468.949	153706.74	1.54%

For the evaluation of a stock, there are two categories of indicators: one is trading indicators, such as the most common opening index, closing index, maximum index, minimum index, and volume, as well as various technical indicators derived from the above, mainly reflecting the trend of current market capital flows and the trend of the indexes brought resulting from capital flows; the other is financial indicators, which are extracted from fundamental factors. Financial indicators are generally

used to analyze the basic situation of industries, enterprises, and national macroeconomics, but many of them are non-quantifiable.

Since we use the neural network for analysis and require quantitative analysis, the indicators used are trading class, and 22 technical indicators such as RSI, KDJ, and MACD are calculated based on the essential indicators already available in the sample. The selected indicators are briefly discussed below.

Relative Strength Index (RSI): RSI is used to analyze the level of market sentiment over a certain period by comparing the average closing gain and average closing loss over some time.

Stochastic indicator (KDJ): KDJ is designed to study the relationship between the highest price, the lowest price, and the closing price and can study the market quickly and intuitively. It consists of the K line, the D line, and the J line.

Moving Average Convergence Divergence (MACD): MACD is designed to reveal changes in the strength, direction, momentum, and duration of a trend in a stock's price.

Directional Movement Index (DMI): DMI is assisted in analyzing the change in the equilibrium point of power between buyers and sellers during the rise and fall of stock prices. It combines directional indicators (+DI and -DI), the ADX and the ADXR.

Deviation rate (BIAS): BIAS reflects the extent to which the price deviates from its moving average over a certain period of time, and from this, the likelihood that the price will maintain its trend can be derived.

On-Balance-Volume (OBV): OBV quantifies the volume and makes a trend line to match the trend line of the stock price, inferring the market atmosphere from the price movement and the volume increase or decrease relationship.

Commodity Channel Index (CCI): CCI is designed to measure whether a stock, forex, or precious metal trade is out of its normal distribution, emphasizing the importance of the average absolute deviation of stock prices in technical analysis.

Rate of Change (ROC): ROC detects the strength of supply and demand forces in advance by calculating the percentage change in the closing price over a certain period of time and then analyzes whether the stock price trend tends to turn.

Bollinger Bands (BOLL): BOLL is based on the statistical principle of standard deviation and confidence interval, using the upper, middle, and lower rail lines to indicate the safety interval of the stock price and thus determine the future trend of the stock price.

In summary, 28 indicators were selected for the final study, with a sample size of 1168, and all technical indicators were calculated through the TA-Lib library.

Table 2. Table of trading indicators

Classification	Description	Abbreviation	Serial Number
Basic Indicators	Opening Index	Open	I1
	Closing Index	Close	I2
	Highest Index	High	I3
	Lowest Index	Low	I4
	Volume	Volume	I5
	Turnover Rate	Change	I6
Technical Indicators	6-Day Relative Strength Indicator	RSI_6	I7
	12-Day Relative Strength Indicator	RSI_12	I8
	24-Day Relative Strength Indicator	RSI_24	I9
	Stochastic Indicator	KDJ_K	I10
		KDJ_D	I11
		KDJ_J	I12
	Moving Average Convergence Divergence	MACD_HIST	I13
		MACD_DIFF	I14
		MACD_DEA	I15
	Directional Movement Index	DMI_+DI	I16
		DMI_-DI	I17
		DMI_ADX	I18
		DMI_ADXR	I19
	6-Day Deviation Rate	BIAS_6	I20
	12-Day Deviation Rate	BIAS_12	I21
	24-Day Deviation Rate	BIAS_24	I22
	On-Balance Volume	OBV	I23
	Commodity Channel Index	CCI	I24
	Rate of Change	ROC	I25
	Bollinger Bands	BOLL_UPPER	I26
		BOLL_MIDDLE	I27
		BOLL_LOWER	I28

3. Data Preprocessing

3.1 Determination of Prediction Targets and Labels

Because the model used in this study is LSTM, a supervised learning approach is adopted. We first divide the samples into training and test sets according to 7:3 and then label the samples. By analyzing the financial time series obtained earlier and combining it with the objective of quantitative strategy, we hope that the model can predict the future stock trend, i.e., up or down, from the available data. It is generally believed that the large increases or decreases in stock prices in the market make more sense for investment purposes, so we group small increases or decreases into one category.

We use the data of the first 60 trading days to predict the cumulative increase or decrease of the index after the next 10 trading days. Let the opening index of the nth trading be $Open_n$, then the closing index of the 10th trading day is $Close_{n+10}$, and the cumulative increase or decrease is

$$C_{n10} = \frac{Close_{n+10} - Open_n}{Open_n} \times 100\% \quad (1)$$

The sample of the *n*th trading day Y_n is labeled as in

$$Y_n = \begin{cases} 0 & C_{n10} < -3\% \\ 1 & -3\% \le C_{n10} \le 3\% \\ 2 & C_{n10} > 3\% \end{cases} \quad (2)$$

The three types of labels correspond to the following meanings: "0" means "big drop," "1" means "small drop or small increase," and "2" means "big increase."

3.2 Standardization

To improve the training and prediction accuracy of the model, the data needs to be normalized before being fed into the model. In this paper, the StandardScaler method of the sci-kit-learn library is chosen to standardize the data of 28 features. This method not only transforms data of different scales into the same scale, making the data directly comparable with each other but also reduces the influence of extreme values in the data beyond the normal range.

3.3 Feature Screening Based on Random Forest

Random Forest is a comprehensive machine learning method that combines random resampling and node random partitioning techniques to construct multiple sub-decision trees. The final output of the model can be obtained based on voting on the output of each sub-decision tree. In the following, the classification accuracy of the random forest classifier is used as the feature separability criterion, and the feature importance ranking is based on the variable importance metric of the random forest algorithm itself.

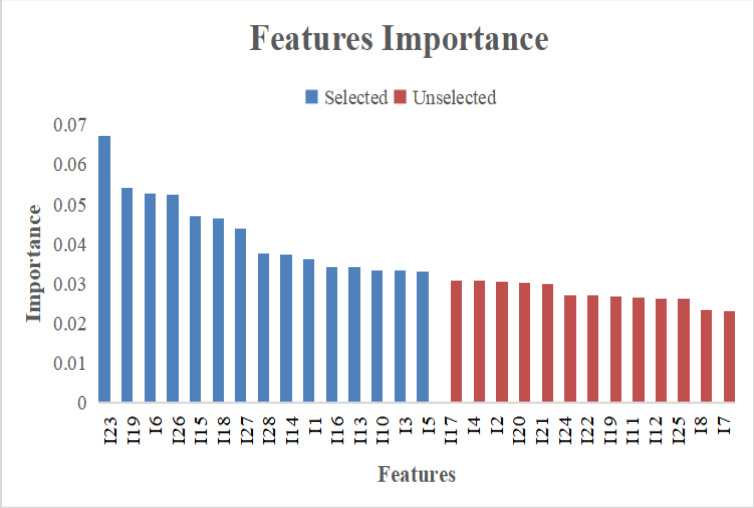


Fig. 1 Importance of each feature.

It is considered the meaning of the features and the degree of importance. The top 15 features of importance are selected for the follow-up study in this paper.

3.4 Principal Component Analysis (PCA)

When the dimensionality of the data is too high, the available data becomes sparse. Especially in financial time series, if the dimensionality is high due to factors such as the range of values taken, then many samples are needed for training, but this is not allowed by objective conditions and is very wasteful of computational resources. Therefore, we use dimensionality reduction for the samples, which can retain most of the information on top of the original data and meet the needs of the sample size and model.

PCA is a statistical method commonly used for dimensionality reduction and denoising. Its basic idea is to analyze the correlation of the original data indicators through the method of linear combination and to form new indicators by combining the indicators with strong correlation so that the new indicators can retain the original information on the one hand, and be uncorrelated with each

other on the other hand, to realize parameter dimensionality reduction. The calculation process of PCA is as in Fig. 2.

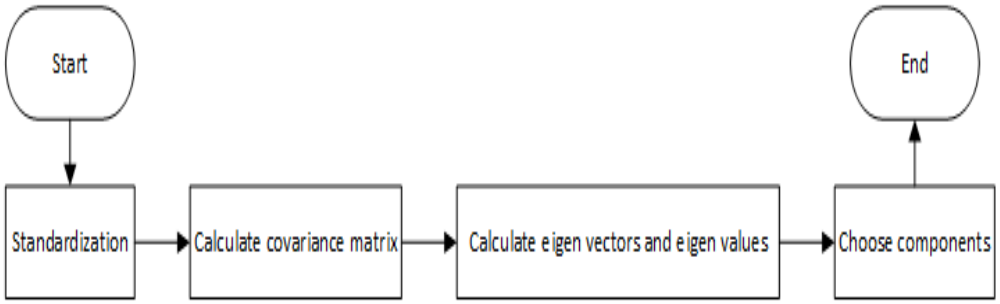


Fig. 2. Flow chart of PCA.

In this paper, PCA is performed on the 15 features screened out. Based on retaining 99% of the information content, the component matrix is obtained as in Table Ⅲ.

Table 3. Component Matrix

	Components								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
I23	0.793	-0.336	-0.118	0.176	-0.034	-0.231	0.291	-0.254	0.088
I19	0.155	0.333	0.812	0.341	-0.112	0.137	0.019	0.009	0.157
I6	0.049	0.221	-0.2	0.472	0.823	0.088	0.012	0.012	0.033
I26	0.925	-0.34	0.06	0.095	-0.019	-0.015	-0.019	-0.004	-0.041
I15	0.604	0.474	0.192	-0.56	0.172	0.136	0.036	-0.054	0.071
I18	0.214	0.397	0.787	0.317	-0.091	0.024	0.079	-0.007	-0.167
I27	0.917	-0.381	-0.011	0.061	-0.017	0.069	-0.056	0.037	-0.023
I28	0.875	-0.414	-0.089	0.02	-0.014	0.16	-0.096	0.081	-0.002
I14	0.586	0.657	0.005	-0.428	0.071	0.118	0.114	0.022	0.082
I1	0.967	-0.195	-0.092	0.05	-0.063	0.074	-0.011	0.065	-0.017
I16	0.479	0.783	-0.222	-0.053	0.051	-0.024	0.023	-0.091	-0.248
I13	0.065	0.656	-0.537	0.294	-0.276	-0.03	0.247	0.223	0.047
I10	0.191	0.664	-0.436	0.324	-0.286	0.145	-0.278	-0.189	0.08
I3	0.971	-0.186	-0.088	0.067	-0.032	0.057	-0.012	0.064	-0.013
I5	0.641	0.468	0.189	-0.037	0.098	-0.512	-0.219	0.094	0.063

4. Models

4.1 The Basic Theory of the Model

Long Short-Term Memory Network (LSTM) is a temporal recurrent neural network specially designed to solve the long-term dependency problem of general RNN (recurrent neural network) with a chained form of repeating neural network modules.

A common LSTM unit comprises a cell, an input gate, an output gate, and a forget gate. The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information into and out of the cell.

Input Gate: The input gate determines how much of the input data of the network at the current moment needs to be saved to the cell state.

Forget Gate: Forget gate determines how much of the cell state needs to be kept to the current moment from the last moment.

Output Gate: The output gate controls how much of the current cell state needs to be output to the current output value.

These gate structures update the cell state by the following recursive equations while activating the mapping from input to output

$$i_t = \sigma(W_i \times [h_{t-1}, X_t] + b_i) \tag{3}$$

$$f_t = \sigma(W_f \times [h_{t-1}, X_t] + b_f) \tag{4}$$

$$o_t = \sigma(W_o \times [h_{t-1}, X_t] + b_o) \tag{5}$$

Where i_t, f_t, and o_t denote the input gate, forget gate, and output gate at the moment $t-1$. σ denotes the activation function. W and b denote the weight matrix and bias, respectively. h_{t-1} denotes the output of the LSTM cell at moment $t-1$, and X_t denotes the input at the moment t.

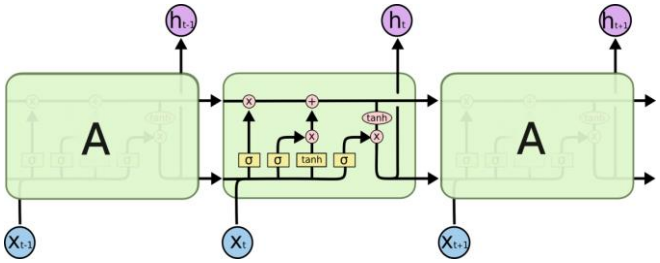


Fig. 3. Structure of the LSTM cell.

In summary, since this paper's index rise and fall prediction problem is a typical time-series problem, i.e., the value at one moment is influenced by the previous moment or several moments, the LSTM model is chosen for training and prediction.

4.2 Model Structure Design

We use Keras, a deep learning framework, to build an LSTM neural network model, and we need to adjust the input sample format. The input to the model is a three-dimensional tensor, whose three dimensions are the number of samples, the time step, and the number of features. The number of samples equals the number of trading days in the training set. The time step, i.e., the number of lags, is also known as the sliding window length. In this dimension, each stock's data in this time step are interconnected, and the appropriate value is 60 days obtained by grid search. The number of features will vary depending on the subsequent model input features.

As for the input and output layers of the model, the number of neurons in the input layer is the number of features in the input sample. The output layer is a fully-connected layer with three neurons, which is the number of categories of cumulative index increase or decrease after ten trading days. The activation function of the output layer is "softmax," the purpose of which is to convert the output into the probability of belonging to each category, and the category with the highest probability is the prediction result. The loss function is "sparse_categorical_crossentropy", which is common for multi-Classification problems.

As for the evaluation metrics of the model, the default metric "Accuracy" is more applicable to the evaluation of multi-classification models. Accuracy measures the proportion of correct classifications of the model. Let $\hat{y}_i$ be the predicted classification of the *i*th sample and y_i be the true classification of the *i*th sample, then the metric on all n samples is

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n} 1(\hat{y}_i = y_i) \tag{6}$$

Where $1(x)$ indicates that the value takes 1 when the predicted result is identical to the true result, and 0 in other cases.

As for the number of hidden layers, there is no universally accepted method for selecting them. From the consideration of the number of features and the size of the dataset, we constructed a model structure with three hidden layers in our study. The activation functions were also compared, and the

best performing "tanh" was selected. The number of neurons in the hidden layer is directly related to the complexity of the solution problem and largely determines the accuracy of the prediction results. If it is too small, more accurate results will not be output. However, if it is too large, the model's generalization ability will become worse and more time-consuming. As it is expected that the model may be overfitted, dropout and L2 regularization are introduced to optimize the model.

Other hyperparameters such as batch size, epoch, optimizer, and learning rate are selected by combining the Bayesian optimization method of keras_tuner library and grid search, and the most effective set of hyperparameters are listed in Table Ⅳ.

Table 4. Model Hyperparameters

Number of neurons in the hidden layer	Optimizer	Learning Rate	dropout factor	L2 regularization factor	batch size	epoch
64, 128, 150	RMSprop	0.001	0.2	0.003	128	80

4.3 Contrast Models

Based on the evaluation criteria established earlier, this paper establishes several comparison models from the training sample aspect to study the model accuracy to establish the optimal prediction model.

We focus on the effect of applying different feature engineering methods to the input samples on the model and performing only feature screening or PCA on the samples or PCA after feature screening reduces the number of features in the original samples. We process the training samples with different feature engineering methods to obtain three models and compare them with the models without feature engineering methods to arrive at the optimal solution.

Table 5. Contrast Models

Models	Feature Engineering Methods
M1	None, with a feature count of 28
M2	Random forest-based screening of the top 15 features in importance, with a feature count of 15
M3	PCA with 0.95 information retained, with a feature count of 9
M4	Random forest-based screening of the top 15 features in importance, and then PCA with 0.99 information retained, with a feature count of 8

The flow of the algorithm of the model is as in Fig. 4.

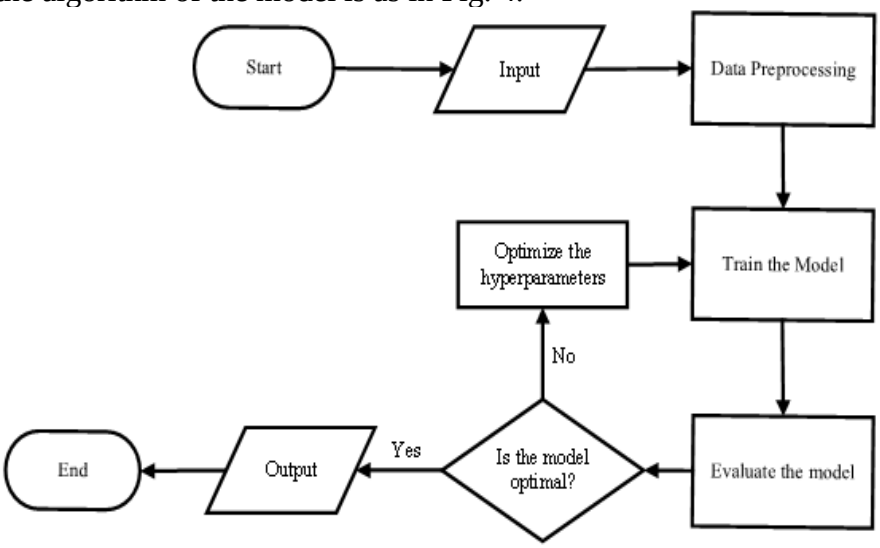
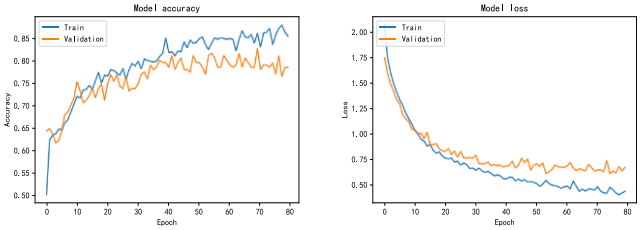


Fig. 4. Algorithm flow chart.

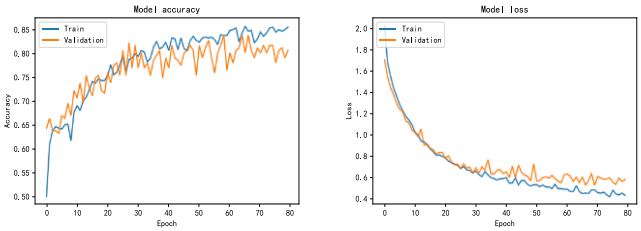
5. Results Analysis

5.1 Performance Test

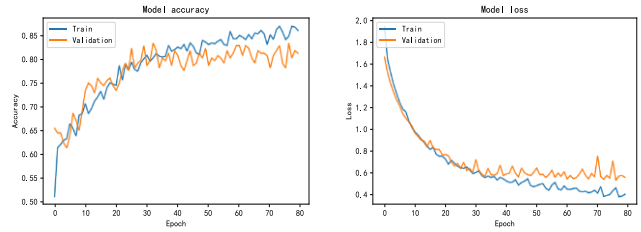
We divide 25% of the samples from the training set as the validation set, then it observes and compares the changes in accuracy and loss of the four models on the training and validation. It is set as the number of epochs increases and finally evaluates the models based on their performance on the test set in a comprehensive manner.



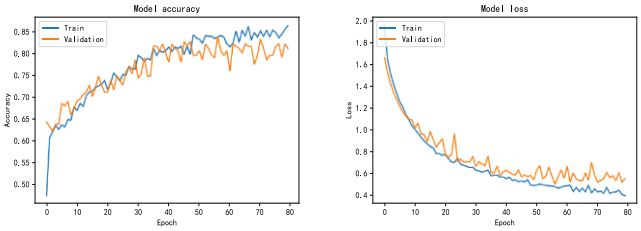
(a) Accuracy curve of M1. (b) Loss curve of M1.



(c) Accuracy curve of M2. (d) Loss curve of M2.



(e) Accuracy curve of M3. (f) Loss curve of M3.



(g) Accuracy curve of M4. (h) Loss curve of M4.

Fig. 5. The Accuracy and Loss.

The results of each model on the test set are as in Table Ⅵ.

Table 6. Accuracy of Models

Models	M1	M2	M3	M4
Accuracy	73%	76%	74%	78%

From Fig. 5(b, d, f, h), it can be seen that the loss tends to level off when it drops to a certain level, indicating that overfitting exists in all four models, and the combination of different methods cannot completely avoid this situation. Comparing the lowest points of the curves in Fig. 5(b, d, f, h), we find that Fig. 5(b) is slightly higher than the other three, and Fig. 5(b) shows overfitting earlier. In addition, comparing Fig. 5(a, c, e, g), the fluctuation of the curve of Fig. 5(a) and the deviation degree

of the two curves are the largest, which indicates that the feature engineering approach to the input samples can reduce the overfitting phenomenon to some extent. Comparing M2, M3, and M4, the curves of the validation set of M4 and the training set are the most similar, while there is no significant difference between M2 and M3.

From Table Ⅵ, it can be derived that the accuracy relationship is in M1 < M3 < M2 < M4.

It shows that the feature engineering approach to input data can improve model prediction accuracy while maintaining a certain amount of information.

In summary, deep neural networks are more sensitive to the information content of the input. Theoretically, the larger the number of features, the more laws the model can learn from, but it is also more likely to have an overfitting phenomenon. Therefore, when the dimensionality of the input data is high, the model accuracy and stability can be improved by screening or dimensionality reduction of the sample features as long as the information content is appropriately maintained. Also, screening the features of the samples before PCA will make the dimensionality reduction more effective.

For the evaluation of each model, in addition to accuracy, other evaluation metrics are introduced. The performance of each of the four models on the test set is investigated in Table Ⅶ.

Table 7. Performance of Models

	M1			**M 2**			**M3**			**M4**		
Labels	0	1	2	0	1	2	0	1	2	0	1	2
Precision	0.63	0.79	0.60	0.71	0.81	0.62	0.76	0.81	0.56	0.79	0.82	0.65
Recall	0.54	0.81	0.61	0.54	0.83	0.67	0.46	0.79	0.71	0.54	0.85	0.69
F1-Score	0.58	0.80	0.61	0.61	0.82	0.64	0.58	0.80	0.67	0.64	0.84	0.67
Quantity	41	219	70	41	219	70	41	219	70	41	219	70

It is observed that model M4 outperforms the remaining three models in all metrics, proving the superiority of combining the two feature engineering methods proposed in this paper over a single one. In addition, it is noted that the metrics of label 1 are all significantly better than those of the other two labels, considering that there is a certain degree of imbalance in the samples in training, i.e., fewer samples correspond to a "big drop" (0) and "big increase," while more samples correspond to "small drop or small increase" (1). It is also consistent with the trading market return trend pattern so that the model will have some bias in the training learning process.

Next, we need to explore whether the accuracy of the model is confounded by other factors, such as policy shocks. We use the outbreak of COVID-19 in 2020 as an example.

The outbreak of COVID-19 was undoubtedly a "black swan" for China's economy, bringing a major impact on consumer spending and corporate production and investment. The fermentation of COVID-19 in the capital market started on January 20. According to relevant news reports, COVID-19 will depress the stock market from both earnings and risk appetite, increasing stock market volatility. On February 3, the first trading day after the Chinese New Year break, the three major stock indices of the A-share market opened sharply lower and maintained a low level of volatility throughout the day. By the end of the day, the Shanghai Composite Index, the Shenzhen Stock Index, and the Growth Enterprise Market Index fell 7.72%, 8.45%, and 6.85%, respectively, with over 3,000 stocks falling in both markets, which is the biggest drop seen in five years [22]. We isolate the 90-day portion of the results after January 20 as in Table Ⅷ, and find that each metric performs poorly. We also infer from this that some unexpected events that lead to a downturn in the financial markets will reduce the overall forecast accuracy.

Table 8. Performance on the 90-Day Portion After Jan. 20

Labels	0	1	2
Precision	0.00	0.60	0.45
Recall	0.00	0.67	0.17
F1-Score	0.00	0.63	0.24
Quantity	5	55	30
Accuracy		0.47	

5.2 Robustness Test

The performance of a predictive model on different test sets reflects the applicability and stability of this model, which is the purpose of our robustness test. For financial markets, the most unpredictable and unwanted phenomenon for investors is a stock market crash. There have been several notable stock market crashes in the history of the Chinese stock market, and the closest one to the time frame of the data set chosen for this paper occurred in 2015. To ensure the time correlation between the samples, we selected the data of the CSI 300 in 2015 as the test set for the robustness test. We did the same preprocessing on the test set and then fed it directly into the model for prediction, and the results are shown in Table Ⅸ.

Table 9. Performance on the 2015 Test Set

Labels	0	1	2
Precision	0.53	0.34	0.56
Recall	0.44	0.66	0.23
F1-Score	0.48	0.45	0.32
Quantity	48	35	40
Accuracy		0.43	

It can be seen that the performance of each metric decreases significantly, indicating the potential instability of the model. Fig. 6 also shows a clear difference in the index's volatility between 2015 and 2016 to 2020, with the former being more unpredictable than the latter, as evidenced by longer and larger continuous rises or continuous falls.

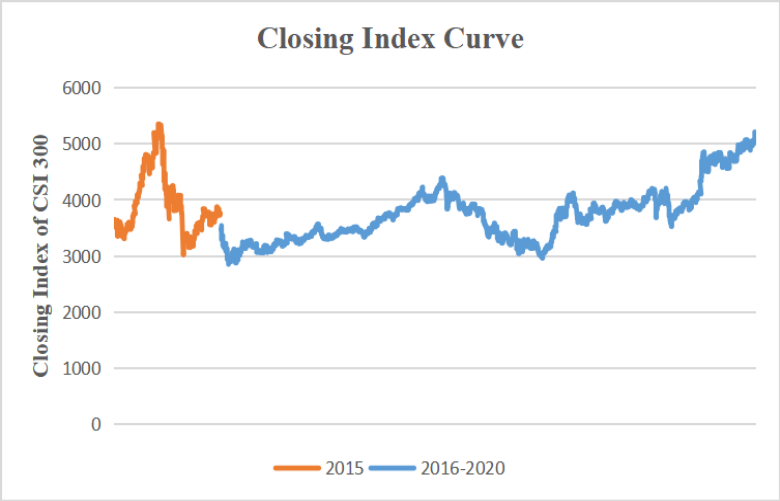


Fig. 6. Closing Index Curve of CSI 300.

It can be seen that the performance of each metric decreases significantly, indicating the potential instability of the model. Fig. 6 also shows a clear difference in the index's volatility between 2015 and 2016 to 2020, with the former being more unpredictable than the latter, as evidenced by longer and larger continuous rises or continuous falls.

After testing, it is concluded that the model proposed in this paper does not make good predictions for such phenomena as stock market crashes or bull markets. However, the training set also does not

include samples with the above phenomena, and the model does not learn the relevant laws, so the results are reasonable.

6. Comparison of LSTM and Other Machine Learning Models

We mainly use the deep learning model in machine learning, i.e., neural network. Of course, many more classical models of machine learning, such as Random Forest, SVM, KNN, are all commonly used supervised learning models. In order to better evaluate the effect of using LSTM in this paper, a comparison experiment was done with the above three models. The three machine learning models are invoked through the sci-kit-learn library, and the parameters are taken as optimal using grid search, and the models are as in Table Ⅹ.

Table 10. Parameters of Machine Learning Models

Random Forest			**SVM**			**KNN**		
n_estimators	***oob_score***	***criterion***	***C***	***gamma***	***kernel***	***n_neighbors***	***p***	***weights***
50	False	gini	10	0.1	rbf	4	1	distance

The performance of the three machine learning models on the test set for each metric is examined in Table Ⅺ.

Table 11. Performance of Machine Learning Models

	Random Forest			**SVM**			**KNN**		
Labels	0	1	2	0	1	2	0	1	2
Precision	0.54	0.74	0.63	0.64	0.77	0.64	0.67	0.82	0.63
Recall	0.33	0.91	0.31	0.56	0.90	0.35	0.67	0.86	0.53
F1-Score	0.41	0.81	0.41	0.60	0.83	0.45	0.67	0.84	0.58
Quantity	45	231	72	45	231	72	45	231	72
Accuracy		0.71			0.74			0.77	

It may be related to the fact that the machine learning model is simple than the neural network and thus has a worse ability to fit the nonlinear data. However, the sensitivity of the model to different samples is not yet determined, so the machine learning model is also affected more when there are "outliers."

In summary, although traditional machine learning methods outperform neural networks in terms of computational complexity and resource consumption, the latter's improvement in accuracy is still significant. For the problem studied in this paper, neural networks are still more effective than traditional machine learning methods when time and hardware conditions allow.

7. Conclusion

In this paper, an LSTM model was established using the Keras framework, which was trained to predict the CSI 300 with a multi-classification prediction, and the accuracy of predicting the range of increase and decrease after ten days reached 78%. We first organized and analyzed the data and formulated the forecasting objectives according to the demand, which was divided into three categories: "big drop," "small drop or small increase," and "big increase." Then, we combined two feature engineering methods to optimize the input samples fully and proposed first to use the random forest to screen out 15 more important features for model prediction to reduce data redundancy and information interference, and then perform principal component analysis on this basis. The last predictions are analyzed in comparison with the no-processing case and the single-processing case, and it is concluded that the method proposed in this paper can achieve more satisfactory results with superiority. In addition, the hyperparameters of the model are also obtained by combining three methods selected from grid search, random search, and Bayesian optimization, which makes the

model more capable of handling nonlinear data and improves the final prediction accuracy to 78%, and makes the results more favorable than those obtained in other studies in the same field. Finally, the results of this paper are compared with those using traditional machine learning models, and the results show that the former obtains better results with the same training set and test set.

The LSTM model with some quantitative indicators as inputs effectively predicts time-series data like stock indices, but it does not entirely exclude other noises, such as some policy shocks and force majeure factors, etc. The inadequacy of choosing only technical indicators is also reflected in the model's prediction results for the index during COVID-19 and the 2015 stock market crash. The accuracy of the model may be further improved if more valid information is added, such as quantifying the policy and news data as the input to the model. In addition, the output of the variable from the model in this paper is up and downrange, which is an intuitive company performance, but we can combine it with value investment to guide us in long-term investment.

References

[1] Bollerslev, T., Marrone, J., Xu, L., & Zhou, H. (2014). Stock Return Predictability and Variance Risk Premia: Statistical Inference and International Evidence. Journal of Financial and Quantitative Analysis, 49(3), 633-661. doi:10.1017/S0022109014000453.

[2] J.H. Kim, A. Shamsuddin, K.-P. Lim Stock return predictability and the adaptive markets hypothesis: Evidence from century-long us data Journal of Empirical Finance, 18 (5) (2011), pp. 868-879.

[3] Murphy JJ (1999) Technical analysis of the financial markets: A comprehensive guide to trading methods and applications. Penguin.

[4] Rao T S, Gabr M M.An introduction to bispectral analysis and bilinear time series models[M].New York: Springer, 2012.

[5] Zheng T, Farrish J, Kitterlin M.Performance trends of hotels and casino hotels through the recession: an ARIMA with intervention analysis of stock indices[J].Journal of Hospitality Marketing & Management, 2016, 25(1): 49-68.

[6] Rangel-Gonzalez J A, Frausto-Solis J, González-Barbosa J J, et al.Comparative study of ARIMA methods for forecasting time series of the mexican stock exchange[J]. Studies in Computational Intelligence, 2018, 749: 475-485.

[7] White H (1988) Economic prediction using neural networks: the case of IBM daily stock returns. In: IEEE Int Conf. Neural Networks. 451–458.

[8] Pan H, Tilakaratne C, Yearwood J (2003) Predicting the Australian stock market index using neural networks exploiting dynamical swings and intermarket influences. In: Australas. Jt. Conf. Artif. Intell. 327–338.

[9] G. Armano, M. Marchesi, A. Murru,A hybrid genetic-neural architecture for stock indexes forecasting,Information Sciences,Volume 170, Issue 1,2005, Pages 3-33,ISSN 0020-0255,https://doi.org/10.1016/j.ins.2003.03.023.

[10] Enke, David & Mehdiyev, Nijat. (2014). A Hybrid Neuro-fuzzy Model to Forecast Inflation. Procedia Computer Science. 36. 10.1016/j.procs.2014.09.088.

[11] M. Khashei , M. Bijari A novel hybridization of artificial neural networks and arima models for time series forecasting Applied Soft Computing, 11 (2) (2011), pp. 2664-2675.

[12] Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. Neural Comput 18:1527-1554.

[13] Nicolas Le Roux, Yoshua Bengio; Deep Belief Networks Are Compact Universal Approximators. Neural Comput 2010; 22 (8): 2192–2207. doi: https://doi.org/10.1162/neco.2010.08-09-1081.

[14] Sutskever I, Hinton GE. Deep, narrow sigmoid belief networks are universal approximators. Neural Comput. 2008 Nov;20(11):2629-36. doi: 10.1162/neco.2008.12-07-661. PMID: 18533819.

[15] Ashwin Siripurapu. Convolutional Networks for Stock Trading[R] . Stanford,2016.

[16] ZHANG Gui-yong. Research on the Application of Improved Convolutional Neural Network in Financial Forecasting[D]. Zhengzhou University, 2016.

[17] HU Yue. Stock market timing model based on Convolutional Neural Network --- Shanghai Composite Index as an example[J]. Financial Economics, 2018(4).

[18] SUN Rui-qi,.Research on U.S. stock index price trend prediction model based on LSTM Neural Network [D]. Capital University of Economics and Business, 2016.

[19] PENG Yan, LIU Yu-hong, ZHANG Rong-fen. Modeling and Analysis of Stock Price Forecast Based on LSTM,[J]. Computer Engineering and Applications, 2019, 55 (11): 209-212.

[20] ZENG An, NIE Wen-jun. Stock Recommendation Based on Deep Bidirectional LSTM[J]. Computer Science, 2019, 46 (10): 84-89.

[21] Kim T, Kim H Y. Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data[J]. PLoS One, 2019, 14(2): e0212320.

[22] DUAN Yu-yuan. The impact of COVID-19 on China's stock market--an empirical analysis based on the pharmaceutical industry[J]. China Business Journal,2020(18):28-30.DOI:10.19699/j.cnki.issn2096-0298.2020.18.028