Article

2024 International Conference on Education, Economics, Management, and Social Sciences (EMSS 2024)

Stock Price Prediction and Investment Simulation - Based on SSA-CNN-LSTM Method

Shangyu Yan [1, *]

[1] Weatherhead School of Management, Case Western Reverse University, Cleveland, USA
[*] Correspondence: Shangyu Yan, Weatherhead School of Management, Case Western Reverse University, Cleveland, USA

Abstract: Stock price fluctuations, which are time-series in nature. At the same time, based on the machine learning Long Short-Term Memory Network (LSTM) has excellent processing ability in predicting long time series, this paper proposes a stock price prediction method using CNN-LSTM optimized for LSTM hyperparameters using Sparrow algorithm. The data used in this study covers a total of 3403 trading days from March 1, 2010 to March 29, 2024 and about 40 technical indicator factors were selected. CNN model was firstly used to extract features from the data. And later, CNN-LSTM network using SSA for parameter optimization uses the extracted features for stock price prediction and simulated trading based on the predicted up and down signals. The experimental results show that the SSA-CNN-LSTM network is able to provide better prediction accuracy than the CNN-LSTM network for price derivation. This prediction method not only provides a better operation idea for actual trading, but also provides practical experience for scholars to study time series in finance.

Keywords: stock predict; technical analysis; deep learning

Published: 01 October 2024

1. Introduction

The trend of stock prices is considered to be an important issue in the economic field, and how to predict the stock price model more accurately is the direction of scholars' efforts [1]. The factors affecting stock prices are also complex, and in general, stock prices are influenced by factors such as the economic environment, the international situation, the industry, the company's financial results and ESG reports, and the development of the stock market [2, 3].

The main ways of analyzing stock prices are divided into fundamental and technical analysis [4]. Generally, fundamental analysis starts from analyzing the intrinsic value of a stock and analyzes the stock price in terms of exchange rates, inflation, financial situation, industrial situation and other factors. Technical analysis, on the other hand, focuses more on the various indicators in the "math game" of stocks. Technical analysis focuses on stock price, trading volume, investor sentiment, etc. It analyzes the trajectory of a stock's price using K charts, technical factors, and other methods. The above two approaches are still the most popular methods of stock analysis.

Technical analysis can well circumvent the shortcomings of fundamental analysis in which the relevant data are updated infrequently, difficult to obtain, and dependent on the analyst's experience. Focusing on technical analysis, its evolution has gone from traditional statistical modeling to incorporating machine learning. The stock market is essentially a dynamic, non-stationary, noisy, chaotic system [5]. Olivas, E.S has been introduced into the field of finance and has played an active role in the valuation of stock prices [6]. The neural networks can be categorized into shallow neural networks and deep learning networks [7]. Deep learning can receive data containing more varied and comprehensive information to further improve the fit due to less restriction on the form of input variables. In 1997, based on the RNN (Recurrent neural network) model, the Long short-term memory (LSTM) model was proposed [8]. Using its characteristic of "time memory", it can pass on the previous time series information very well, and it can also use correlation to select the characteristics that can have an impact on the stock price, and by constantly adjusting the parameters to obtain the appropriate model, thus increasing the accuracy of the prediction [9]. In practice, it has been found that the LSTM model takes a long time to compute and is easily affected by the parameter settings. Attempts have been made to combine other models with the LSTM model to maximize its advantages. In order to reduce the amount of input data, researchers have tried to use Principal Component Analysis (PCA) in conjunction with LSTM models and have made some progress. Along with the further popularity of machine learning, there are also attempts to apply Random Forest (RF) with LSTM model for machine learning, and the prediction accuracy is further improved compared with PCA by utilizing the feature extraction and importance assessment ability of RF model, which is confirmed in the back test [10]. And in order to better extract the temporal and spatial features of time series species, researchers tried to use convolutional neural network (CNN)-LSTM hybrid model for stock price prediction [11, 12]. Comparing the hybrid model with RNN and single LSTM model, the CNN-LSTM model showed superior results in terms of statistical metrics [11]. Another idea is to introduce an evolutionary algorithm, which is not easy to fall into the local optimum, to find the optimal LSTM parameters, so as to improve the prediction ability of the model. Researchers have used genetic algorithm (GA) to optimize the time window and topology of LSTM networks, which significantly improves the performance of prediction models [13]. The classical optimization algorithms include ant colony algorithm, particle swarm optimization algorithm and so on.

In this paper, we will use the sparrow optimization algorithm to set the hyperparameters of the LSTM model. The sparrow algorithm analogizes the process of finding parameters to that of a sparrow foraging for food to data to avoid falling into a local optimum [14]. The main contribution is proposing a stock price forecasting model using the Sparrow Algorithm (SSA) to optimize CNN-LSTM network. We use three stocks in different industries as experimental subjects. Firstly, Random Forest is utilized to find out the factors that have a significant impact on stock price, and then, LSTM, CNN-LSTM and SSA-CNN-LSTM models are utilized to predict the stock price and compare them respectively. Finally, the SSA-CNN-LSTM model price prediction rise and fall is used as a signal and the holding buy and sell in the cycle is simulated, and it is found that SSA-CNN-LSTM achieves a more desirable positive return on the simulated investment in BEIDAHUANG GROUP. This finding further proves the reliability of stock technology investment, and the reasonable algorithm setting further reduces the threshold of stock learning for investors, so that individual investors can have summarized reference conclusions in the actual trading process.

2. Methodology

2.1. Cnn-Lstm Model

CNN can extract local features by convolution operation, so it is widely used in feature engineering, and the cell structure of LSTM is able to capture the long-term dependencies in the sequence, so it is also widely used in time series.Compared with CNN-LSTM model, it has obvious advantages in the extraction of local features, reducing the dimension of the data, enhancing the robustness, and shortening the computation time. Combining the features of CNN and LSTM, more and more scholars try to predict stocks based on CNN-LSTM.

2.2. SSA

The sparrow algorithm is inspired by sparrow feeding behavior in nature - sparrows as a group of organisms have different roles for different individuals when searching for food [14]. When a detector succeeds in obtaining "food", it will cause other individuals to "follow" it. Functionally, detectors provide the population with foraging directions and areas while followers follow detectors in search of food. The monitors, in turn, keep a close eye on the foraging location and abandon the food if they detect danger. By constantly updating their location, sparrows are able to optimize resources during the foraging process.

The position of detectors is updated as follows(1):

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^{t} * \exp\left[-i/(a * iter_{max})\right], \ R_2 < ST \\ x_{i,j}^{t} + Q * L, \ R_2 \geq ST \end{cases} \tag{1}$$

This formula is an updated formula for predator position.ST denotes the safety value and R2 is the warning value. When the value < ST it means it is safe. At this time the predator search range is large, when the value ≥ ST, it means that there are too many predators in the area and need to move to a safe place.

The position of followers would follow the formula shown in (2):

$$x_{i,j}^{t+1} = \begin{cases} Q * \exp\left(\frac{x_{worst}^{t} - x_{i,j}^{t}}{i^2}\right), \ if \ i > 2/n \\ x_p^{t+1} + \left|x_{i,j}^{t} - x_p^{t+1}\right| * A^{+} * L, \ otherwise \end{cases} \tag{2}$$

This formula represents the update of the joiner position

The anti-predator behavior of the sparrow population is expressed through the following(3), Vigilante positions are updated by this formula:

$$x_{i,j}^{t+1} = \begin{cases} x_{best}^{t} + \beta * \left|x_{i,j}^{t} - x_{best}^{t}\right|, \ if \ \ f_i > f_g \\ x_{i,j}^{t} + K * \left(\frac{\left|x_{i,j}^{t} - x_{worst}^{t}\right|}{(f_i - f_w) + \varepsilon}\right), \ if \ f_i = f_g \end{cases} \tag{3}$$

2.3. Convolutional Neural Network

Often abbreviated as "CNN"， was proposed by Lecun et al. in 1998[15]. CNN is a class of neural networks that contain convolutional computation and have deep structure, which is one of the representative algorithms for deep learning. We apply it to processing one-dimensional data (stock price) [16]. Briefly, in terms of feature extraction, CNN captures and inputs data features through a small number of parameters, which are recombined to form high-level data features. These new features will be used in the fully connected layer for the next step of regression or classification prediction. We use a one-dimensional convolutional neural network. The CNN structure typically consists of an input layer, a convolutional layer, a pooling layer, a fully connected layer and an output layer.

Typically, a convolutional layer can be represented by the following equation (6):

$$x_j^l = f\left(\sum_{i=1}^{l} x_i^{l-1} \otimes k_{i,j}^l + b_j^l\right) \tag{4}$$

where x_j^l is the output feature map of the j neuron of the current layer (l layer); l is the number of input features of the lth convolutional layer; x_i^{l-1} is the output feature map of the previous layer (layer l-1), and x_i^{l-1} is also the input of the current layer l; ⊗ denotes the convolutional operation; k_{ij}^l denotes the convolution kernel of the i neuron of the j neuron from the l-1 to the l layer; b_j^l is the standard deviation of the jth neuron of layer l; f is the activation function, which is obtained using the following equation (7):

$$f(x) = \frac{1}{1+e^{-x}} \tag{5}$$

2.4. Long And Short-Term Memory Network

Long Short-Term Memory (Long Short-Term Memory) is a temporal recurrent neural network (Recurrent Neural Network) that has the ability to memorize over time. The structure of the network consists of one or more units with forgettable and memorizable functions. It was proposed in 1997 [17] to solve the vanishing gradient problem over back-propagation-over-time in traditional RNNs (Recurrent Neural Networks). The important components include Forget Gate, Input Gate, and Output Gate, which are responsible for deciding whether the current input is adopted or not, whether it is memorized for a long time or not, and whether the memorized input is output or not in the current time or not, respectively. While the Gated Recurrent Unit has a memorizati on function, LSTM is often used in data and scenarios with time-series characteristics, especially when observing samples over a long period of time.

Here is the formula and process of LSTM:

The forgetting gate mainly determines the ratio of the input feature data X and the data features h of the previous hidden layer to be retained/discarded by the model, and the forgetting gate formula(8) is shown below:

$$f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f) \tag{6}$$

The input gate primarily controls the extent to which the input feature data X and the results of the current model's operations are output to the internal memory unit. The output gate formula(9) is shown below:

$$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i) \tag{7}$$

The output gate controls how much feature information the model outputs based on the current degree size of the internal memory cell, and the output gate formula (10-11) is shown below:

$$O_t = \sigma(U_O x_t + W_O h_{t-1} + b_O) \tag{8}$$

$$h_t = o_t * \tanh{(c_t)} \tag{9}$$

The internal memory unit determines the importance of the input information and its formula(12-13) is as follows:

$$u_{t =} \tanh{(U_u x_t + W_u h_{t-1} + b_u)} \tag{10}$$

$$C_t = f_t * c_{t-1} + i_t * u_t \tag{11}$$

where i is the input gate, f is the forgetting gate, and o is the output gate, this indicates that at time t, when X_t denotes the output data and h_t hides the position, U and W are the matrix weights. The symbol "*" denotes the outer product of vectors and "+" denotes the superposition operation.

Here is a schematic of the flow of the LSTM network (Figure 1):

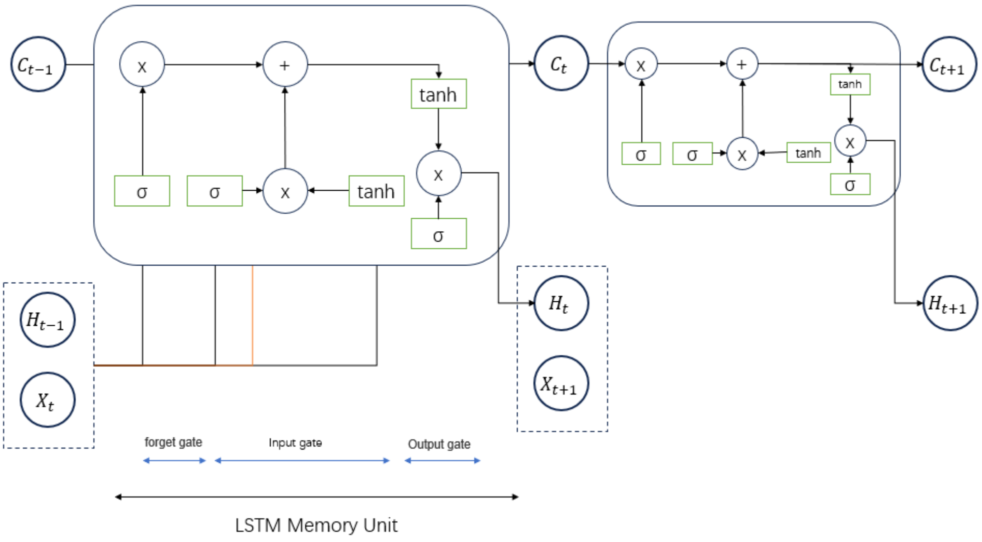

Figure 1. The Schematic of the Flow of the LSTM Network.

In LSTM model, the oblivion gate, input gate, and output gate have their own (U,W,B),and these parameters are also obtained during the training process. When the value of the forgetting gate f is close to 1 and the output gate i is close to 0, then the feature information of the previous layer is not saved and the model still memorizes the previous feature information. When f is close and i is close to 1, the useful information in the previous input features is preserved and what was memorized by the model before is forgotten. Thus, the individual units in the LSTM model jointly determine the output result h.

2.5. SSA-CNN-LSTM

The model utilizes the SSA model for optimization for the following 5 parameters in LSTM: number of neurons, dropout, batch_size for 3-layer LSTM network. while the CNN network part is not modified.

3. Experimental

To demonstrate the effectiveness of SSA-CNN-LSTM, we compare it with LSTM,CNN-LSTM in the same operating environment. According to the influencing factors, besides the basic volume, opening price, closing price, high price, low price, there are some technical factors and their descriptions are given in the appendix.

Here, we show some of the data using China Merchants Bank as an example (Table 1):

Table 1. Performance Comparison of SSA-CNN-LSTM, LSTM, and CNN-LSTM.

trade_date	close	open	high	low	change	pct_change	vol
20100301	16.3	15.95	16.35	15.9	0.4	2.52	813009.3
20100302	16.46	16.4	16.55	16.3	0.16	0.98	1097808
20100303	16.49	16.46	16.5	16.25	0.03	0.18	709333.9

3.1. DATA

3.1.1. DATA SOURCES

In this experiment, China Merchants Bank (600036), Zhuhai Huafa Group Co., Ltd. (600325), and BEIDAHUANG GROUP (600598) are selected as the experimental data, and the daily trading data for the period from March 1, 2010 to March 29, 2024 ,the daily trading data of 3368 trading days are obtained from the tushare database. Each data contains the basic factors of opening price, high price, low price, closing price, volume, and technical factors. Some of the data is shown in the table below. The data of the first 2694 trading days are used as the training set and the data of the last 674 trading days are used as the test set.

The technical indicator factors used are:

MACD,KDJ,RSI,BOLL_UPPER/MID/LOWER,CCI,PE,PS,DV_RATIO,TOTAL_SHARE,FLOTE_SHARE,FREE_SHARE,TOTAL_MV,SMA,MA,AD,ADOSC,OBV,ADX,APO,BOP,CMO

3.1.2. Data Preprocessing

Although all of the data come from the tushare database, they still have some problems. Here, we address the missing values, derive the filled values by weighted average of the prices in the days before and after, standardize the format of the data to float64, and handle the renaming of the titles with inconsistent names.

In terms of data standardization, the MAX-MIN method was used to scale all input attributes to the [0,1] interval to avoid interference from extreme values, which is formulated as follows:

$$x^*=\frac{x_i-x_{min}}{x_{max}-x_{min}} \tag{12}$$

3.2. Model Evaluation

In this model, we use four indicators commonly used in statistics to evaluate the model fit, adaptability. They are root mean square error(RMSE), mean absolute error(MAE), Mean Absolute Percentage Error(MAPE) and R-square(R^2).

The RMSE is the square root of the mean of the squared prediction error, which is very sensitive to large errors and is often used to explain which data points the model underperforms. The formula for calculating RMSE is as follows:

$$RMSE = (\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2)^{\frac{1}{2}} \quad (13)$$

Where $\widehat{y_i}$ is the predictive value and the y_i is the true value. The smaller of the value of RMSE , the better the forecasting.

The MAE is the average of the absolute errors between all individual observations and the predicted values it directly reflects the average absolute difference between the predicted and actual values directly. The formula for MAE is as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|(y_i - \hat{y}_i)| \quad (14)$$

Where $\widehat{y_i}$ is the predictive value and the y_i is the true value. The smaller of the value of MAE, the better the forecasting.

The Mean Absolute Percentage Error (MAPE) is commonly used to evaluate the performance of prediction methods. For prediction methods in the field of machine learning, MAPE is also a measure of prediction accuracy, which is usually expressed as a percentage of precision.

The formula for MAPE is as follows:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{\widehat{y_i}-y_i}{y_i}| \quad (15)$$

Where $\widehat{y_i}$ is the predictive value and the y_i is the true value.

The R^2 calculation formula is as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\widehat{y_i}-y_i)^2}{\sum_{i=1}^{n}(\overline{y_i}-y_i)^2} \quad (16)$$

3.3. Implementat of the Model

There are three models in this paper: LSTM, CNN-LSTM, and SSA-CNN-LSTM. in terms of LSTM hyperparameters, except for the hyperparameters of the SSA-CNN-LSTM model which are obtained by the parameter seeking of the SSA algorithm, the remaining two models are of the same setup, and the model parameters are specified in the following table:

For CNN-LSTM model (LSTM follow the same parameters):

For CNN part: Convolution layer filters = 128 , Convolution layer kernel_size = 3 , Convolution layer activation function = relu , Pooling layer pool_size = 1 ,

For LSTM part: The number of LSTM neurons in each of the 3 layers is 128 64 32 , activation function = tanh ,Time_step = 12 , Batch_size = 32 Learning_rate = 0.001 , Optimizer = Adam , Loss function = MSE , Epochs = 10

For SSA part: M=10, POP=2, dim=5, P_Percent=0.2 , ST=0.8

POP is the population size of sparrows, M is the number of iterations, ST is the safety threshold, P_Percent is the ratio of the percentage of finders in the sparrow population, and the number of parameters to be optimized for the optimization of the optimization-seeking dimension generation step.

4. RESULT

4.1. Descriptive Statistics

The technical factors were assessed for importance by the Random Forest algorithm, and the following is the order of importance in descending order, with the ranking ending when the cumulative importance exceeds 90%.

Shown in Figure 2-4 are China Merchants Bank, Zhuhai Huafa Group Co., Ltd., BEIDAHUANG Group:

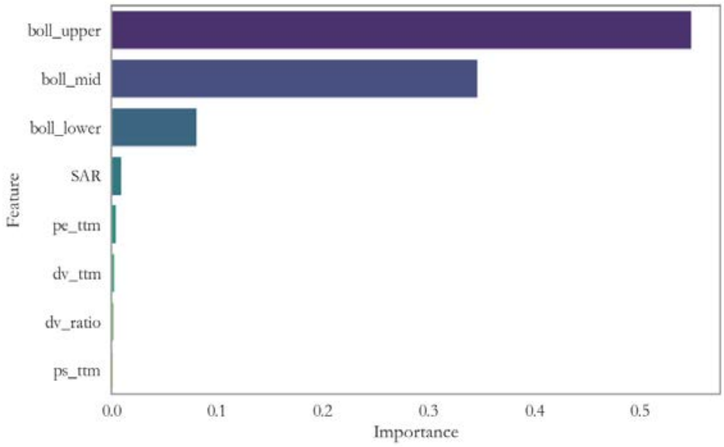

Figure 2. China Merchants Bank.

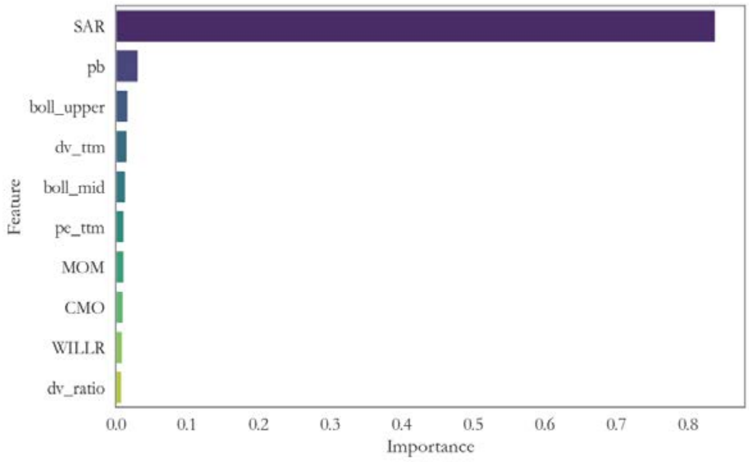

Figure 3. Zhuhai Huafa Group Co., Ltd.

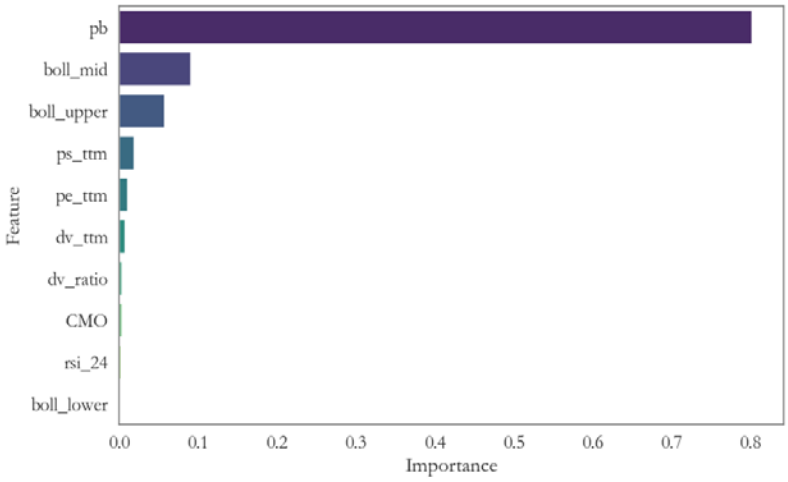

Figure 4. BEIDAHUANG GROUP.

Here, we can find different technical indicator factors for stocks in different industries that can better estimate their price fluctuations. But combining these three charts we find that the boll line, price-to-earnings (pe), price-to-book (pb), and price-to-sales (ps) ratios are always predictive potential factors.

4.2. Compared Based on the Evaluation

We show the predictive ability of the three models in the T+1 ,T+3, T+6 timeframe, categorized by company in Table 2-4.

Table 2. China Merchants Bank.

Evaluation	T+1		T+3		T+6	
Index	LSTM	CNN-LSTM	LSTM	CNN-LSTM	LSTM	CNN-LSTM
RMSE	3.4447	1.8962	4.0355	2.1528	4.9172	2.9634
MAE	2.176	1.3864	2.7217	1.6067	3.1062	2.2652
MAPE	0.0489	0.0339	0.061	0.0396	0.0725	0.055
R^2	0.8257	0.9472	0.7617	0.9322	0.6477	0.872

Table 3. Zhuhai Huafa Group Co., Ltd.

Evaluation	T+1		T+3		T+6	
Index	LSTM	CNN-LSTM	LSTM	CNN-LSTM	LSTM	CNN-LSTM
RMSE	0.5505	0.4126	0.6348	0.6050	0.9842	1.0234
MAE	0.4735	0.3169	0.5015	0.4623	0.7765	0.8488
MAPE	0.0662	0.0409	0.0659	0.0595	0.1005	0.1140
R^2	0.8926	0.9397	0.8571	0.8702	0.6566	0.6288

Table 4. BDH Group.

Evaluation	T+1		T+3		T+6	
Index	LSTM	CNN-LSTM	LSTM	CNN-LSTM	LSTM	CNN-LSTM
RMSE	0.7072	0.5974	0.9417	0.8448	1.5940	1.3668
MAE	0.5504	0.4456	0.6783	0.6049	1.2482	1.0630
MAPE	0.0366	0.0294	0.0446	0.0398	0.0836	0.0712
R^2	0.8744	0.9104	0.7760	0.8197	0.3474	0.5201

4.3. SSA Optimization Results

We use the sparrow search algorithm to perform parameter optimization based on the parameters set in the previous section for the number of neurons, batch_size, and dropout in the three layers of the LSTM network, and the optimization process is shown in BEIDAHUANG Group in Figure 5 as an example:

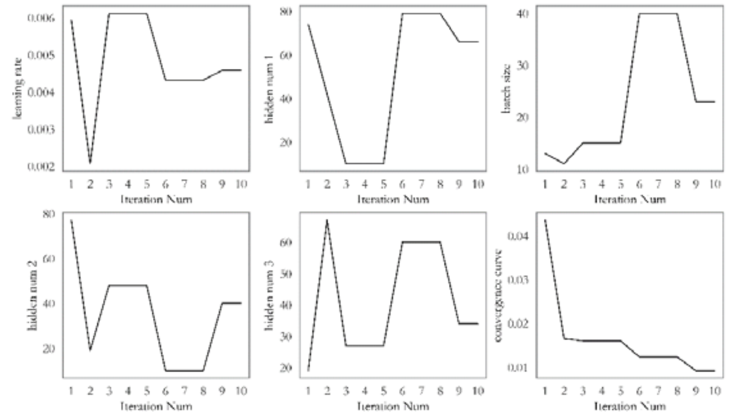

Figure 5. Parameter Optimization Process for LSTM Network Using Sparrow Search Algorithm (BEIDAHUANG Group).

Ultimately, we identified this set of hyperparameters as the optimization result of the LSTM:

Learning_Rate =0.00458. The number of neurons in each of the three layers is 66 40 34.Batch_size = 23

As a comparison, we similarly present the prediction results(Model 3) of the CNN-LSTM(Model 2) networks in parallel, see the following Table 5:

Table 5. The Prediction Results of the CNN-LSTM Networks.

Evaluation	**T+1**		**T+3**		**T+6**	
Index	**Model 2**	**Model 3**	**Model 2**	**Model 3**	**Model 2**	**Model 3**
RMSE	0.5974	0.5884	0.8448	0.6469	1.3668	0.9850
MAE	0.4456	0.4654	0.6049	0.4743	1.0630	0.8073
MAPE	0.0294	0.0314	0.0398	0.0316	0.0712	0.0554
R^2	0.9104	0.9131	0.8197	0.8943	0.5201	0.7508

And is the Sparrow algorithm a "one and done" option? Is it possible to carry over the hyperparameters from one search to other stocks in the same cycle? Here the authors use the China Merchants Bank stock as an example, bringing in the results of the search parameters in the BDH Group stock, and the final prediction results of China Merchants Bank are as follows (Table 6):

Table 6. The Final Prediction Results of China Merchants Bank.

Evaluation	**T+1**		**T+3**		**T+6**	
Index	**Model 2**	**Model 3**	**Model 2**	**Model 3**	**Model 2**	**Model 3**
RMSE	1.8962	2.9465	2.1528	2.5424	2.9634	3.7274
MAE	1.3864	2.1333	1.6067	1.9379	2.2652	2.7591
MAPE	0.0339	0.0493	0.0396	0.0472	0.0550	0.0640
R^2	0.9472	0.8725	0.9322	0.9054	0.8720	0.7976

4.4. Summative Comparison

Here, we take Heilongjiang Beidahuang Group of State Farms and Land Reclamation as an example to compare the prediction results against LSTM, CNN-LSTM, SSA-CNN-LSTM networks at three time points T+1 T+2 T+3 demonstrated in Figure 6.

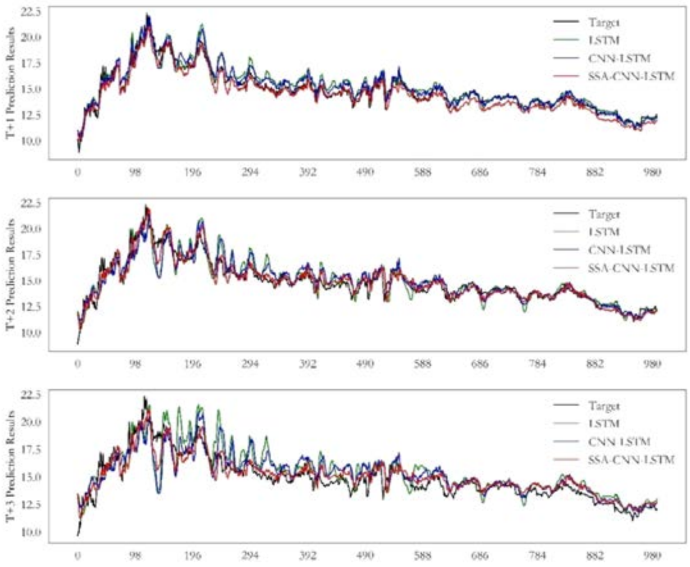

Figure 6. Prediction Comparison for LSTM, CNN-LSTM, and SSA-CNN-LSTM at T+1, T+2, T+3 (Heilongjiang Beidahuang Group).

The CNN-LSTM network optimized by the Sparrow Search algorithm has better performance in price prediction. When the prediction period is extended, the gap between its prediction ability and that of CNN-LSTM network will be further widened. As we can see from the graph, the prediction gap is not only in the extremes but also in the incorrect estimation of the upward or downward trend of the stock - a fatal error in real trading.

When looking at stock prices during the T+3 period, we see that all three neural networks have extremely large errors on trading days 98-196 - in fact, stock prices are relatively stable during this period, but all three models judge that there is a sharp drop at this time. Combined with the previous data, we find that this trend already exists - and that the errors are further amplified as the estimation time increases. This suggests that, in addition to further considering the spatio-temporal dimensions of the time series and optimizing the network's hyperparameter settings, we should also pay attention to the type and quality of sampling of data factors. Without good data, it is difficult for deep learning networks to evaluate good prediction results.

4.5. Investment Simulation

In order to verify the reference value of utilizing the model, this paper designs the following trading strategy, signaling the price increase/decrease on the second day of the model's prediction:

Strategy 1:

1. is based on the model's prediction that the stock price is 1% above the current five-day SMA; and

2. is where the divergence value DIF in the MACD breaks upwards through the short-term average of the divergence value DEA and both are greater than zero.

Strategy 2: Always Hold

In this paper, the prediction period is set to be the same as the stock date estimation period, and the initial capital is set to 100.The trading strategy based on SSA-CNN-LSTM model is called Strategy 1, which targets BDH Group. The comparison of the strategies for the two stocks is presented in Table 7 below:

Table 7. Comparison of Trading Strategies.

	Strategy 1	**Strategy 2**
Sharpe ratio	1.204312	0.32808
Annualized Compound Yield(%)	26.0651	5.5572
Maximum withdrawal(%)	15.3557	50.6278
Total Asset(ten thousand)	247.7404	123.5933
Total Revenue(%)	147.7404	23.5933

In the meantime, we'll show the line graph in Figure 7 at the same time - it shows us the results of the change in assets over time for both strategies.

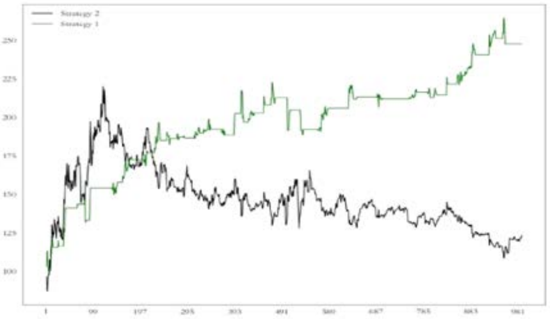

Figure 7. The Results of the Change in Assets.

Over the statistical period, we find that Strategy 1 does not consistently outperform Simple Holding Strategy 2. But over the entire statistical period, it achieves quite good results: the strategy's Sharpe Ratio ends up at 1.20, compared to Simple Holding's 0.32. Its annualized return ends up at 26%, five times that of the Simple Holding strategy. The maximum retracement was 15.35%, a reduction of about 35%. Total assets doubled to $2.47 million, nearly twice as much as the Simple Hold. It ended up with a total return of 147.74%, 120% higher than the simple holding strategy.

5. Conclusion

In this paper, a stock price prediction model based on SSA-CNN-LSTM is proposed for the financial time series forecasting problem. First, the model analyzes the importance of features and identifies the technical factors that have the greatest impact on stock prices. Then, the CNN-LSTM model is utilized to better consider the feature factors included in the time series and the SSA model is used to find better hyperparameter settings for the LSTM to improve the prediction accuracy of the model. The results show that SSA-CNN-LSTM has the best performance in the evaluation metrics set in the paper. This paper also conducts simulated investment tests on the prediction results. The results show that the trading strategy based on the SSA-CNN-LSTM model is able to obtain better returns and effectively control the maximum retracement compared to simply holding stocks. This indicates that under the premise of fully utilizing the deep network, the model has a certain predictive ability for the rise and fall of stocks, which can be used as a reference for predicting the index trend or trading practice.

Meanwhile, since the stocks used in this study are from three different industries, they will present different results in the final comparison. In the long run, a clear direction is to focus the stocks studied in the model on the same industry and observe whether different types of stocks have factor screening sets and parameter settings that are more suitable for their industries; and also, to add textual and sentiment factor analysis to include more information when considering the factors that affect stock prices.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

1. Fama, E. F. (1995). Random walks in stock market prices. Financial analysts journal, 51(1), 75-80.
2. Zhang, L. L., & Kim, H. (2020). The influence of financial service characteristics on use intention through customer satisfaction with mobile fintech. Journal of System and Management Sciences, 10(2), 82-94.
3. Badea, L., Ionescu, V., & Guzun, A. A. (2019). What is the causal relationship between stoxx europe 600 sectors? But between large firms and small firms?. Economic Computation & Economic Cybernetics Studies & Research, 53(3).
4. Park, C. H., & Irwin, S. H. (2007). What do we know about the profitability of technical analysis?. Journal of Economic surveys, 21(4), 786-826.
5. Ruo-yu, Q. I. A. O. (2019). Stock Prediction Model Based on Neural Network. Operations Research and Management Science, 28(10), 132.
6. Olivas, E. S., Guerrero, J. D. M., Martinez-Sober, M., Magdalena-Benedito, J. R., & Serrano, L. (Eds.). (2009). Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques: Algorithms, methods, and techniques. IGI global.
7. Asteris, P. G., & Mokos, V. G. (2020). Concrete compressive strength using artificial neural networks. Neural Computing and Applications, 32(15), 11807-11826.
8. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
9. Long, J., Chen, Z., He, W., Wu, T., & Ren, J. (2020). An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market. Applied Soft Computing, 91, 106205.
10. Ma, Y., Han, R., & Fu, X. (2019, October). Stock prediction based on random forest and LSTM neural network. In 2019 19th International Conference on Control, Automation and Systems (ICCAS) (pp. 126-130). IEEE.
11. Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-based model to forecast stock prices. Complexity, 2020(1), 6622927.
12. Jin, G., & Kwon, O. (2021). Impact of chart image characteristics on stock price prediction with a convolutional neural network. Plos one, 16(6), e025 3121.

13. Chung, H., & Shin, K. S. (2018). Genetic algorithm-optimized long short-term memory network for stock market prediction. Sustainability, 10(10), 3765.
14. Xue, J., & Shen, B. (2020). A novel swarm intelligence optimization approach: sparrow search algorithm. Systems Science & Control Engineering, 8(1), 22-34.
15. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
16. Kuang, D., & Xu, B. (2018). Predicting kinetic triplets using a 1d convolutional neural network. Thermochimica acta, 669, 8-15.
17. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of SOAP and/or the editor(s). SOAP and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.