PREDICTION OF THE STOCK ADJUSTED CLOSING PRICE BASED ON IMPROVED PSO-LSTM NEURAL NETWORK

YU-XUAN LUO, YI JI

School of Electrical and Information Engineering, Jiangsu University, Zhenjiang, Jiangsu 212013, China
E-MAIL: 2222107005@stmail.ujs.edu.cn, jiyi@ujs.edu.cn

Abstract:

Volatility in the stock market has a significant impact on all finance-related fields. As an important part of stock data, the adjusted closing price often reflects the attention of market funds to a stock, helping predict the market movement of the next trading day, especially for short-term investors. With the development of artificial intelligence technology, the machine learning algorithms are widely applied to predict stock trends. However, the noisy, nonlinear, and chaotic nature of stock price changes makes the prediction not accurate enough. Hence, we proposed a hybrid prediction model combining improved particle swarm optimization (IPSO) and long short-term memory (LSTM) neural network to predict the adjusted closing price of the stock. In this paper, nonlinear methods are presented to optimize the velocity inertia weight and learning factors of traditional particle swarm optimization (PSO). Meanwhile, IPSO is used to optimize the hyperparameters of LSTM neural network to improve its prediction accuracy. The experiments proved that the proposed IPSO-LSTM outperformed the Autoregressive Integrated Moving Average model (ARIMA), LSTM, and PSO-LSTM on the prediction of the S&P 500 Index. Furthermore, the Dow Jones Industrial Average Index (DJI) and Nasdaq Composite Index (IXIC) were chosen to verify the accuracy and robustness of the model we put forward.

Keywords:

Prediction; Particle swarm optimization; Long short-term memory network; The adjusted closing price of the stock

1. Introduction

Predicting stock data has been a hot topic of research for decades [1]. With the rapid development of the Internet, the financial market produces a lot of online stock data every day, such as opening price, closing price, high price, low price, trading volume, adjusted closing price, and so on. However, the dynamic, complex, and chaotic characteristics of the stock market make it difficult to predict stock accurately. Therefore, although rich historical stock information provides data resources for the majority of investors and analysts, it is far from enough to rely on personal subjective judgment to predict stock trends [2].

To objectively predict stock data, researchers used various traditional statistical methods, such as Generalized Auto Regressive Conditional Heteroskedasticity (GARCH), ARIMA, and Multiple Linear Regression Model (MLR). However, these traditional models are proved to be more suitable for linear prediction. Therefore, the prediction accuracy of the traditional linear model on the complex and nonlinear stock data is not credible.

Recently, with the development of machine learning and deep learning methods, such as ANN (Artificial Neural Network), SVM (Support Vector Machine), and RNN (Recurrent Neural Network), improved clustering algorithms and improved neural networks have begun to be widely applied in the field of stock prediction. Although neural networks including RNN are often utilized to predict time series data [3], RNN has the problem of gradient disappearance when dealing with long time series data. However, the time memory unit of the LSTM neural network is capable to learn the long and short-term information of time series, which makes up for the shortcoming of RNN prediction. Therefore, the LSTM is suitable for predicting time series with long time intervals, including stock prediction. Whereas, parameters of the LSTM model are usually manually adjusted through experience, which lacks efficiency and makes the results random and subjective. To solve the above problems, we utilize the IPSO algorithm to automatically adjust the hyperparameters in LSTM, thus improving the efficiency and accuracy of LSTM network.

The contributions of this paper are as follows:

(1) The PSO algorithm is utilized to update the main hyperparameters of the LSTM network adaptively, which can help improve the efficiency and accuracy of the LSTM network prediction.

(2) To solve the local convergence problem, we update the velocity inertia weight and learning factors of particles in a novel nonlinear way. So that the global searching ability of particles is strengthened to avoid particles falling into local optimum easily.

2. Related research theories

2.1. ARIMA model

The full name of ARIMA is the autoregressive integrated moving average. ARIMA is made up of autoregressive and moving average, whose expression is ARIMA(p, d, and q). p represents the autoregressive order, d represents the difference order, and q is the moving average order. ARIMA is a traditional linear time series prediction method with high prediction accuracy [4], so it is selected as the baseline for our research. Details of the process of constructing the ARIMA model for time series prediction can be seen in Shao's paper [5].

2.2. LSTM neural network

LSTM neural network is called long short-term memory neural network which owns neurons that have the capacity for long-term memory. The structure of LSTM neurons can be seen in Figure 1. The working process of every LSTM memory unit can be expressed by the following formulas (1-5):

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \quad (2)$$

$$c_t = f_t c_{t-1} + i_t\, tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (4)$$

$$h_t = o_t\, tanh(c_t) \quad (5)$$

where i, f, c, and o refer to input gate, forget gate, cell status, and output gate, respectively. x_t represents the input of the current moment. h_{t-1} and h_t refer to the output from the previous moment and the current output, respectively. c_{t-1} and c_t refer to the cell status of the previous moment and the current cell status. W and b represent the weights and bias of the corresponding matrix in each part, respectively. Meanwhile, σ and tanh represent sigmoid and hyperbolic tangent activation function, respectively.

The structure of multiple gates allows the LSTM to selectively analyze and utilize the cell state at each moment in the RNN. The number between [0,1] obtained during the operation determines how much information is allowed to pass through the structure. The forget gate determines which part in the previous should be abandoned while the input gate decides which part has access to the current status. Finally, the output gate achieves the current output based on the previous output, newest status, and current input.

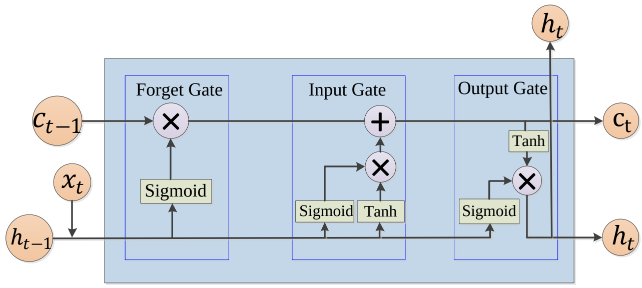


FIGURE 1. The Structure of LSTM

2.3. Particle Swarm Optimization (PSO) algorithm

Particle Swarm Optimization (PSO) is a population-based metaheuristic optimization technology, proposed by Eberhart and Kennedy in 1995 [6]. The basic idea of the particle swarm optimization algorithm is to find the optimal solution through cooperation and information sharing among individuals in the population.

PSO is initialized as a set of random particles, and then the optimal position is found through the iterative motion of the particles. The position and velocity of particle i in n-dimensional space are represented as two vectors, $X_i = (x_1, x_2, \cdots, x_n)$ and $V_i = (v_1, v_2, \cdots, v_n)$, respectively. Each particle has a fitness value computed by the objective function. The particle's best fitness value corresponds to the best position. In each iteration, the particle updates itself by tracking the two best values. The first is the best position for the particle itself, denoted by $pbest_i$. All particles in the population can share information about the obtained best point, so the second is the global best position, denoted by $gbest_i$. After these two optimal values are obtained, the particle updates its velocity and position by formula (6) and formula (7).

$$V_i^{t+1} = \omega \times V_i^t + c_1 \times rand \times \left(pbest_i - X_i^t\right) + c_2 \times rand \times \left(gbest_i - X_i^t\right) \quad (6)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (7)$$

where V_i^t and V_i^{t+1} refer to the velocity vector of particle i at iteration t and t+1. ω is the velocity inertia weight that determines how much of the velocity each particle inherits in the next iteration from the previous iteration. c_1 and c_2 are learning factors that influence the speed at which each particle moves towards its own optimal position and the swarm optimal position, which both equal to the constant 2 in general. $rand$ represents a random number between [0,1]. X_i^t and X_i^{t+1} refer to the position vector of particle i at iteration t and t+1.

2.4. PSO-LSTM model

In this paper, the PSO algorithm is utilized to optimize the hyperparameters of the LSTM neural network so as to improve its prediction efficiency and accuracy. In this study, the position and velocity vector of each particle are both defined as four-dimensional vectors. These four dimensions represent the number of PSO iterations, the number of nodes in the two hidden layers of the LSTM model, and the learning rate of LSTM, respectively. The first three parameters are random integers in the range of [1, 300] and the last one is a random variable between [0.001, 0.1]. c_1, c_2, ω are initialized as values 2, 2, 1, respectively. The number of particles is set as 20 and the maximum number of iterations for PSO is set to 20. Meanwhile, the value of the look-back period in LSTM is set to 15, which means the LSTM network predicts the value of the next day by learning the patterns of the previous 15 days.

3. Proposed algorithm

3.1. IPSO-LSTM model

PSO algorithm often falls into the problem of local optimum in the iterative process [7]. To solve this problem, we proposed a new IPSO algorithm to avoid premature convergence. Specifically, we improved the velocity inertia weight ω and learning factor c_1, c_2 in formula (6).

3.2. Improved velocity inertia weight

The velocity inertia weight is one of the most significant variable parameters in the PSO model [8]. The velocity inertia weight is set as constant 1 in the traditional PSO. However, a large exploration speed may cause particles to miss the global optimal position and then lead PSO to fall into the trap of local optimum. Therefore, we propose a novel nonlinear method to update the velocity inertia weight ω in formula (6) during iteration. The novel inertia weight is given by formula (8):

$$\omega_t = 0.5 + \frac{1}{1+e^{t/t_{max}}} \quad (8)$$

where ω_t is the improved velocity inertia weight of particles in iteration t, whose value represents the ratio of the velocity to be inherited in the next iteration. Meanwhile, t and t_{max} refer to the current and maximum iterations of PSO, respectively. Obviously, the value of ω_t is in the range of [0.5, 1) and is getting smaller over the course of iteration.

This nonlinear method slows down the decreasing rate of ω as the number of iterations increases, which ensures that the individual particle can search more carefully around its initial position. At the same time, the value of ω keeps above 0.5, ensuring the global exploration ability of particles. In general, the improved velocity inertia weight balances the global exploration ability and local exploration ability of particles, which can help avoid PSO falling into local optimization.

3.3. Improved learning factors

c_1 and c_2 in formula (6) are usually set as the constant 2 in traditional PSO. This means that in any iteration particles move at the same speed towards the particle best position and towards the global best position, which may lead to a lack of diversity of particles and an incomplete exploration of the whole n-dimensional space. Hence, we proposed a novel nonlinear method to update the learning factor c_1 and c_2. The novel learning factors are given by formula (9) and formula (10):

$$c_1 = 2.55 - 1.15 \times \sqrt{tanh(\pi t \,/\, t_{max})} \quad (9)$$

$$c_2 = 1.5 + 0.95\sqrt{tanh(\pi t \,/\, t_{max})} \quad (10)$$

where t and t_{max} refer to the current and maximum iterations of PSO, respectively. Obviously, the value of c_1 is getting smaller during iteration while the value of c_2 is getting greater. Meanwhile, the value of c_1 and c_2 are in the range of (1.4, 2.55] and [1.5, 2.45), respectively. The improved learning factors can increase the diversity of the particle swarm at the beginning of the iteration to allow the particles to fully explore the entire region, and gather the particles to explore the global optimal solution at the end of the iteration.

3.4. The proposed IPSO-LSTM framework

In this paper, the LSTM neural network is optimized by the improved PSO algorithm. We then use IPSO-LSTM to predict the next day's adjusted closing price by learning the previous 15 days' adjusted closing price from the historical data of the stock to be predicted. The overall process is shown in Figure 2.

The specific steps of the whole process are listed as follows:

Step 1: The historical adjusted closing price data of the stock to be predicted are proportionally divided into a training set and a test set. Then the two datasets are normalized by the Z-Score standardization method.

Step 2: Initialize the IPSO parameters including the size of the particle swarm, particle dimension, the number of iterations, velocity inertia weight, learning factors, particles' position, and particles' velocity. Initialize the LSTM parameters including the number of nodes in each hidden layer, epoch, batch size, and the value of the look-back period.

Step 3: Train the LSTM model with the training set. Hyperparameters in the LSTM network are automatically updated by the IPSO algorithm.

Step 4: The root mean square error (RMSE) is selected as the fitness value of particles in IPSO. The fitness of each particle is recorded during the iteration, deciding the individual optimal fitness and the global optimal fitness.

Step 5: Particles update their velocity and position by formula (6) and formula (7).

Step 6: The optimal LSTM hyperparameters are recorded when the maximum iteration times t_{max} is reached. Otherwise, return to *Step 3* to continue the iteration.

Step 7: Through the above steps, the IPSO-LTSM model can build the optimal LSTM network for predicting stock data. Finally, the prediction results are evaluated by four specific evaluation indicators.

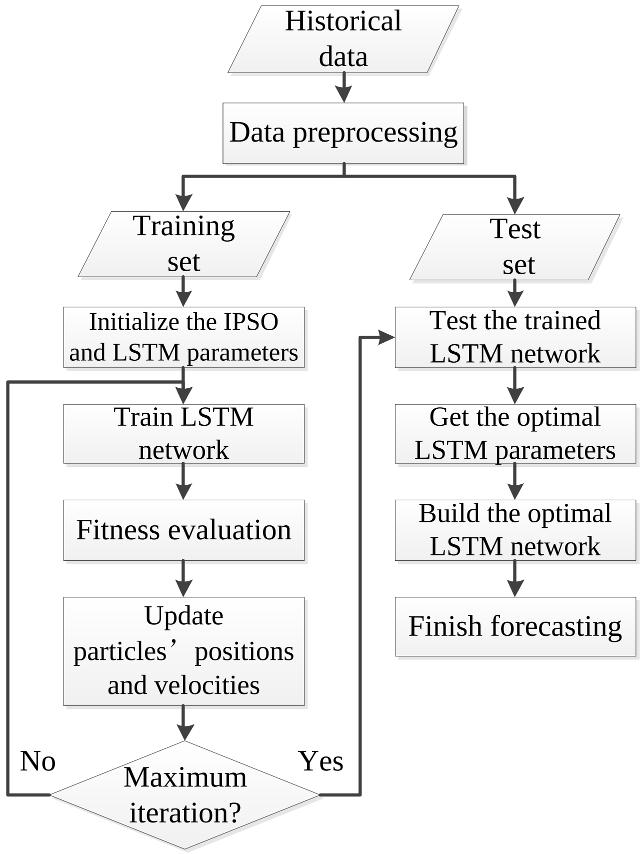


FIGURE 2. Flowchart of the IPSO-LSTM model

4. Experiments

4.1. Development environment

The development environment of this study is Python 3.7, PyCharm Edu 2021.1.2 IDE on a Windows 10 operating system. The overall process is implemented under the TensorFlow machine learning platform.

4.2. Dataset and preprocessing

In this research, we chose the S&P 500 Index which tracks 500 publicly traded companies in the United States as our main object to be predicted. The ticker symbol of S&P 500 Index is ^GSPC. The experimental dataset consists of the daily historical adjusted closing price data in the past 10 years from 1 July 2011 to 29 June 2021. The raw dataset is first divided into two parts, the first 70% and the remaining 30%, which represent the training set and the test set, respectively. The result is shown in Figure 3.

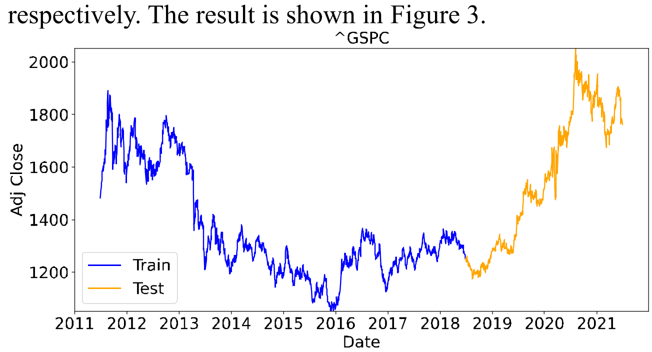


FIGURE 3. The training set and test set of the S&P 500 Index

Since values of the training set and test set are too large to predict, the two datasets are then both standardized to time series with a mean of 0 and a standard deviation of 1 by the Z-Score standardization method, which is expressed as formula (11).

$$x'(t) = \frac{x(t)-\mu}{\sigma} \tag{11}$$

where t refers to every trading day, $x(t)$ refers to the raw adjusted closing price, $x'(t)$ refers to the adjusted closing price after standardization, μ and σ refer to the mean value and the standard deviation of the raw adjusted closing price, respectively.

4.3. Evaluation indicators

In this study, we choose four indicators to evaluate the performance of the prediction model, including root mean square error (RMSE), mean absolute percentage error (MAPE), mean absolute error (MAE), and R2_score (R^2). These four evaluation indicators are expressed by the following formulas (12-15):

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(y(t) - \hat{y}(t)\right)^2} \tag{12}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{y(t)-\hat{y}(t)}{y(t)}\right| \times 100\% \tag{13}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N}|y(t) - \hat{y}(t)| \tag{14}$$

$$\mathrm{R}^2 = 1 - \frac{MSE(y(t)-\hat{y}(t))}{Var(y(t))} \tag{15}$$

where N represents the number of samples in the test set. $y(t)$ and $\hat{y}(t)$ refer to the true value in the test set and the corresponding prediction result. Meanwhile, Var and MSE indicate the variance and mean squared error, respectively.

4.4. Parameter settings of different models

The parameter settings of the three baselines and IPSO-LSTM proposed in this study are listed in Table 1. Where p represents the autoregressive order, d represents the difference order, and q is the moving average order. Node1 and Node2 refer to the number of nodes in the first and second LSTM hidden layer, respectively. η indicates the learning rate of LSTM. The look-back period refers to the number of the past few days LSTM network relies on to predict data for the next day.

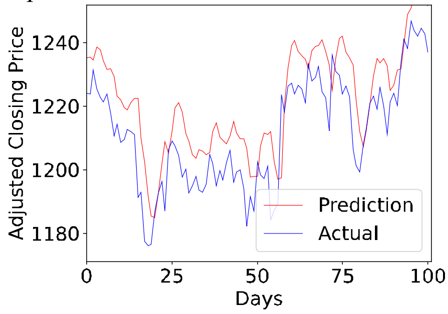
TABLE 1. Parameter settings of four models

Model	Parameter Settings
ARIMA	(p, d, q) = (1, 1, 1)
LSTM	Node1 = 200, Node2 = 200, Epoch = 50, η = 0.001, Batch_size = 64, look-back period = 15
PSO-LSTM	Node1, Node2, Epoch ∈ [1, 300], $\eta \in [0.001, 0.1]$, Batch_size = 64, look-back period = 15, ω = 1, $c_1 = c_2 = 2$
IPSO-LSTM	Node1, Node2, Epoch ∈ [1, 300], $\eta \in [0.001, 0.1]$, Batch_size = 64, look-back period = 15, $\omega_t \in [0.5, 1)$, $c_1 \in (1.4, 2.55]$, c2 ∈ [1.5, 2.45)

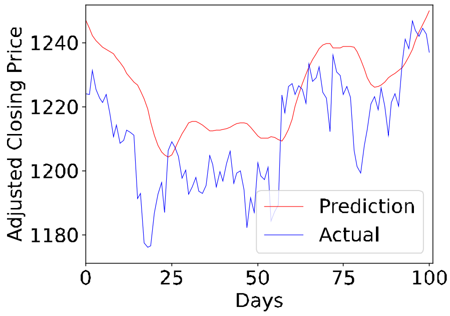
4.5. Comparison of prediction results of different models

In this research, we used ARMIA, LSTM, PSO-LSTM, and IPSO-LSTM respectively to predict the aforementioned S&P 500 Index (^GSPC) data set. Their respective prediction results for the first 100 days of the test set are shown in Figure 4. Meanwhile, we specifically compared the prediction performance of the four models through the four evaluation indicators introduced before. The evaluation results of the whole test set are compared in Table 2.

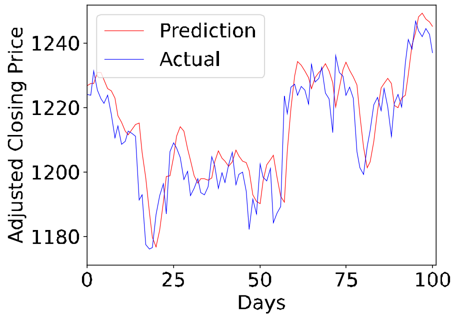
As shown in Table 2, the proposed IPSO-LSTM model outperforms the other three baselines on all four evaluation indicators, which means the predictive performance of our model is superior to the other three models.



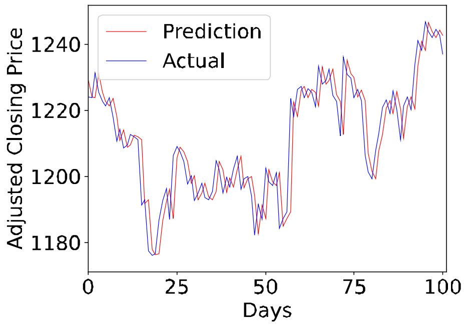
(a) ARIMA



(b) LSTM



(c) PSO-LSTM



(d) IPSO-LSTM

FIGURE 4. Prediction results of four models

TABLE 2. Numerical representation of predictive performance

Model	RMSE	MAPE(%)	MAE	R^2
ARIMA	18.05	0.80	12.70	0.9947
LSTM	30.42	1.33	21.25	0.9848
PSO-LSTM	18.20	0.75	12.03	0.9948
IPSO-LSTM	**16.75**	**0.67**	**11.12**	**0.9958**

4.6. Robustness verification

To further verify the robustness of the proposed IPSO-LSTM model, experiments were carried out on two other representative stock indexes: the Dow Jones Industrial Average Index (DJI) and Nasdaq Composite Index (IXIC). These two stocks are preprocessed and predicted in the same way as the S&P 500 Index. The corresponding prediction results for the first 100 days of these two test sets shown in Figure 5 and evaluation results through four indicators shown in Table 3 indicate the robustness of the proposed IPSO-LSTM model.

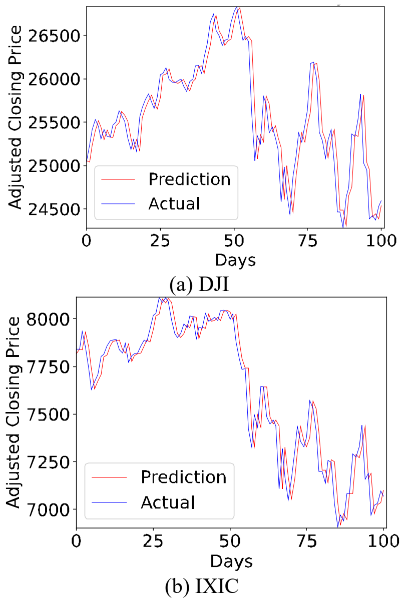


FIGURE 5. Prediction results of DJI and IXIC

TABLE 3. Evaluation results

Stock data	RMSE	MAPE(%)	MAE	R^2
DJI	370.20	0.89	231.82	0.9847
IXIC	146.04	1.06	99.65	0.9959

5. Conclusion

This paper proposed a novel improved PSO-LSTM (IPSO-LSTM) hybrid model to predict the adjusted closing price of any stock. Specifically, nonlinear methods are presented to optimize the velocity inertia weight and learning factors of the particle swarm optimization (PSO) algorithm. Experiments proved that the improved PSO is able to search for better hyperparameters for the LSTM network. We evaluated the performance of the proposed IPSO-LSTM model on three representative stock indexes (S&P 500 Index, DJI, and IXIC) using four evaluation indicators (RMSE, MAPE, MAE, and R^2), and compared it with three baselines (ARIMA, LSTM, and PSO-LSTM). The comparative study indicated that the proposed IPSO-LSTM outperforms the other three models.

References

[1] Z. Su, M. Lu, and D. X. Li, “Financial empirical application of deep learning: Dynamics, contributions and prospects”, J. Financ. Res. Beijing, pp. 111-126, March. 2017.

[2] F. G. Deng and H. L. Wang, “Application of LSTM neural network in prediction on stock price trends, based on research on individual stock data in US and HK stock market”, Financ. Econ. Beijing, pp. 96-96, July. 2018.

[3] J. Li and Y. F. Lin, “Time series data prediction based on multi-time scale RNN”, Comput. Appl. Software Shanghai, Vol. 35, pp. 33-37, 62, October. 2018.

[4] Adebiyi. AA, Adewumi. AO and Ayo. CK, “Stock price prediction using the ARIMA Model”, 16th UKSim-AMSS International Conference on Computer Modelling and Simulation, pp. 106-112, March. 2014.

[5] Shao. BL, Li. ML, Zhao. Y and Bian. GQ, “Nickel price forecast based on the LSTM neural network optimized by the improved PSO algorithm”, Mathematical Problems in Engineering, Vol. 39, September. 2019.

[6] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proc. IEEE Int. Conf. Neural Netw., Perth, WA, Australia, pp. 1942–1948, Nov/Dec. 1995.

[7] S. Sengupta, S. Basak, and R. Peters. "Particle Swarm Optimization: A survey of historical and recent developments with hybridization perspectives." Machine Learning & Knowledge Extraction, pp. 157-191, October. 2018

[8] Ji Y, Liew. AWC, Yang LX, “A novel improved particle swarm optimization with long-short term memory hybrid model for stock indices forecast”, IEEE ACCESS, Vol. 9, pp. 23660-23671, Feb. 2021.