Journal of Global Information Management
Volume 30 • Issue 7

CVaR Prediction Model of the Investment Portfolio Based on the Convolutional Neural Network Facilitates the Risk Management of the Financial Market

Zheng Wu, Shanxi Vocational and Technical College of Finance and Trade, China
Yan Qiao, China University of Mining and Technology, China
Shuai Huang, Shanghai University of Finance and Economics, China
HsienChen Liu, East China Normal University, China*

ABSTRACT

In summary, firstly, a method for establishing a portfolio model is proposed based on the risk management theory of the financial market. Then, a prediction model for CVaR is established based on the convolutional neural network, and the improved particle swarm algorithm is employed to solve the model. The actual data analysis is implemented to prove the feasibility of CVaR prediction model based on deep learning and particle swarm optimization algorithm in financial market risk management. The test results show that the investment portfolio CVaR prediction model based on the convolutional neural network can obtain the optimal solution in the 18th generation at the fastest after using the improved particle swarm algorithm, which is more effective than the traditional algorithm. The CVaR prediction model of the investment portfolio based on the convolutional neural network facilitates the risk management of the financial market.

KEYWORDS

Big Data, Convolutional Neural Network, CVaR Model, Deep Learning, Digital Finance

INTRODUCTION

In recent years, with the continuous advancement of global financial integration, the Internet financial market has gained huge development space; however, the Internet financial market itself is unstable, and the advancement of globalization has caused increasingly frequent fluctuations (Song et al., 2020). Both companies and financial institutions, as well as ordinary investors, are threatened by financial risks in the financial market (Zhang, 2020). The financial risks will have a great impact on the development and operation of both companies and financial institutions, and affect the survival of individuals. Besides, the increasing financial risks and the ultimate explosion will also bring an impact on the national and even the global financial market, thereby seriously endangering the stability and health of economic growth (Tang et al., 2019). what's more, the diverse investment portfolios in the financial market can diversify the financial risk management, leading to the decline in financial risks.

DOI: 10.4018/JGIM.293288
*Corresponding Author

Internationally, there are many researches on the risk management of Internet financial market. Some scholars described VaR, a new indicator to describe the risk of financial market, and applied it to the risk management of China's financial market. The feasibility of VaR was verified according to the actual situation (Almahdi & Yang, 2019). Also, some scholars compared VaR and CVaR, deeming that CVaR can more fully reveal financial risks; applied CVaR to optimize the investment portfolio (Gao & Su, 2020). As a financial risk management tool, the CVaR-based portfolio model has difficulty in solving when its dimension is relatively high (Zhou et al., 2019). Therefore, some scholars applied artificial intelligence (AI) algorithms such as genetic algorithm to solve this model, and this global random search algorithm shows great advantages in dealing with such problems (Fischer & Krauss, 2018). Some scholars applied Particle Swarm Optimization (PSO) to solve the model based on deep learning, so as to establish a more optimized investment portfolio. In order to obtain higher solving efficiency, some scholars enhanced the intelligent algorithm based on deep learning (Bai et al., 2021). Although these researches put forward the portfolio model to deal with the financial risks, the problem of solving the model still needs to be addressed accordingly.

To sum up, firstly, a method for establishing a portfolio model was proposed based on the risk management theory of the financial market in this paper. Then, a prediction model for CVaR was established based on the CNN, and the improved particle swarm algorithm was employed to solve the model. The actual data analysis was implemented to prove the feasibility of CVaR prediction model based on deep learning and PSO algorithm in the financial risk management. Moreover, the proposed algorithm was compared with the traditional particle swarm algorithm. Therefore, this research provides an important reference for the adoption of deep learning methods in the financial risk management.

METHODS

Risk Management Theory of Financial Market

The fundamental and core part of financial risk management is the measurement of financial risks (Chiang et al., 2020; Han et al., 2021). The continuous development of the financial market makes the transaction of financial assets have a larger scale and become more complex (Gong et al., 2019; Niu et al., 2021). Under such circumstances, investors will construct the corresponding portfolio models to balance the asset returns and financial risks. As a result, the selection of risk control indexes in portfolio model has also become a hot topic, and the purpose of choosing risk control indexes is to measure financial risks.

In the current financial risk management, the variance and standard deviation is widely used to measure financial risks (Ghimire et al., 2019). This measurement method is a kind of symmetric measurement in differential application, so the return rate of assets shows a normal distribution (Latchoumi et al., 2019; Wang et al., 2020). The calculation method of variance is shown in Eq. (1), and the calculation method of standard deviation is shown in Eq. (2).

$$V = E\left[\left(R - r\right)^2\right] \tag{1}$$

$$\sigma = \sqrt{E\left[\left(R - r\right)^2\right]} \tag{2}$$

where, R is return rate of assets, and r is expected rate of return. Minimizing variance will reduce the downward offset of asset returns; yet for the upward offset of asset returns, it will also eliminate part of it. Therefore, the variance and standard deviation method is likely to reduce the return on financial assets, and it mainly deals with the risks brought by high probability events in the financial market, and can't effectively measure the risks for some low probability events (Korovkinas et al., 2020; Xiang et al., 2021; Yi, 2021). However, low probability events in the financial market tend to have a huge impact on the return of financial assets and have a severe impact on financial institutions and investors (Faia et al., 2018). In order to measure the financial risk of complex asset portfolio comprehensively, the VaR method was introduced. VaR essentially refers to the maximum possible loss of a financial portfolio in a certain period of time in the future when the market is in normal fluctuation, after given a certain confidence level (Ma et al., 2017), as shown in Eq. (3).

$$\Pr\left(\Delta p > VaR\right) = 1 - a \tag{3}$$

where, Pr represents possibility; confidence level is represented by a; and the loss of a portfolio at a given time is represented by Δp. It is necessary to know the confidence and holding period of the asset portfolio to calculate VaR, and there is a positive correlation between VaR and holding period of the asset. The higher the confidence, the more likely the portfolio is to lose less than the value of VaR in a given period of time.

VaR model can predict the risk of financial market in extreme cases, which facilitates the financial risk management. However, it can't meet the requirements of the consistency axiom of risk measurement, and it is difficult to achieve results in measurement calculation of the dynamic form (Ghasemiyeh et al., 2017). To solve these problems, CVaR based on VaR is proposed. CVaR can satisfy the condition of consistency axiom and has subadditivity and convexity, and the definition is shown in Eq. (4).

$$CVaR = E\left(X \,/\, X \geq VaR\right) \tag{4}$$

where, CVaR is the confidence level, the loss of financial asset portfolio exceeds the average value of VaR (Shi et al., 2017). CVaR and VaR can reflect the potential losses of asset portfolio in the financial market. CVaR also takes the tail of losses into consideration and is more excellent as a risk measurement index.

CVaR-Based Portfolio Model

In order to achieve the corresponding investment objectives, investors can constantly adjust the asset portfolio according to the fluctuations of the financial market when managing the portfolio (Al-Sulttani et al., 2017). There are transaction costs associated with the adjustment of portfolio. Transaction costs refer to the sum of fees paid by investors in transactions (Das & Padhy, 2018). The calculation of transaction costs in practice will use quite complex equations, so it is difficult to implement the calculation of transaction costs in the financial risk management of the portfolio. Traditional portfolio models generally ignore the impact of transaction costs, but the transaction costs affect the return rate of the investment portfolio to a large extent (He et al., 2018). Therefore, the establishment of the CVaR-based portfolio model in this study will consider the impact of transaction costs.

In the case of multiple variables, the expression of CVaR is shown in Eq. (5).

$$CVaR = E\left(\mathrm{f}\left(x\right) / \mathrm{f}\left(x\right) \geq VaR\right) \tag{5}$$

where, $f(x)$ *is* a loss function. CVaR refers to the expected value, so there must be two situations: continuous and discrete (Amirhosseini & Hosseini, 2018). The function of continuous CVaR expected loss is shown in Eq. (6).

$$\psi\left(x,\alpha\right)=\int_{f(x)\leq\alpha}P\left(y\right)dy \tag{6}$$

The calculation of VaR and CVaR is shown in Eqs. (7) and (8).

$$VaR=\min\left\{\alpha\in R:\psi\left(x,\alpha\right)\geq\gamma\right\} \tag{7}$$

$$CVaR=\frac{1}{1-\gamma}\int_{f(x)\geq VaR}P\left(y\right)f\left(x\right)dy \tag{8}$$

Assuming that there are *m* situations in the market in the future and the probability of occurrence is *1/m*, the calculation of discrete CVaR is shown in Eq. (9).

$$CVaR=VaR+\frac{1}{\left(1-\gamma\right)m}\sum\left[f\left(x\right)-VaR,0\right]^{+} \tag{9}$$

It is difficult to calculate continuous and discrete CVaR directly, and VaR is needed to calculate CVaR, which also forms a difficulty, for which the auxiliary functions are introduced to calculate, as shown in Eq. (10).

$$F\left(x,\alpha\right)=\alpha+\frac{1}{1-\gamma}\int_{y\in R}\left[f\left(x,y\right)-\alpha\right]^{+}P\left(y\right)dy \tag{10}$$

where, both *x* and *α* are variables. Eq. (10) is taken as the objective function to obtain the optimal solution, and the result is the global optimal solution, as shown in Eq. (11).

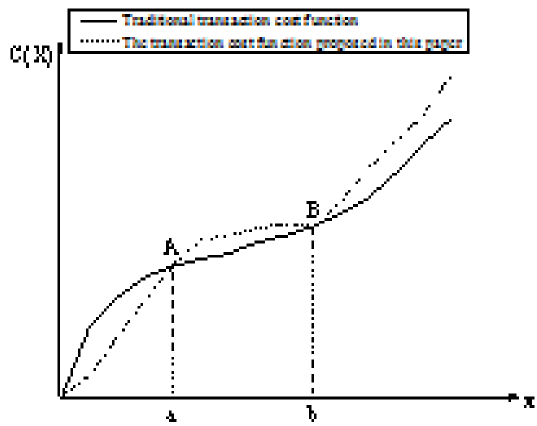
$$CVaR=\min_{y\in R}F\left(x,\alpha\right) \tag{11}$$

The purpose of investors' investment in the financial market is obtaining the maximum return on assets and realize the risk control of the portfolio at the same time of obtaining the return (Maiyar et al., 2019). The return rate of asset portfolio is expressed by Eq. (12).

$$R=\sum_{i=1}^{m}r_{1}r_{i} \tag{12}$$

Since transaction cost has great influence on the income of asset portfolio, the transaction cost is studied. The traditional transaction cost function and the transaction cost function proposed in this study are shown in Fig. 1.

Figure 1. The performance of the traditional transaction cost function and the transaction cost function analyzed in this article



As shown in Figure 1, the traditional transaction cost function takes that the transaction cost function of segment *AB* will show a linear change. This study deems that the transaction volume of asset portfolio is relatively large, so the transaction cost is also relatively high. Therefore, when looking for the optimal investment portfolio, its transaction cost and transaction volume can appear nonlinear increase only, and the transaction cost function *C* is shown in Eq. (13).

$$C = \begin{cases} \rho\sqrt{X} & 0 \le X \le a \\ \rho\left(hX^{2} + gX + s\right) & a \le X \le b \\ \rho\left(kX^{3} + hX^{2}\right) & b \le X \le 1 \end{cases} \tag{13}$$

where, g is the coefficient of the primary term and the coefficient of the quadratic term is represented by h; and the coefficient of the cubic term is expressed in terms of k.

To sum up, the CVaR-based portfolio model in this study is shown in Eq. (14).

$$MinF\left(x,\alpha\right) = \alpha + \frac{1}{\left(1-\gamma\right)m}\sum\left[f\left(x,y\right) - \alpha\right]^{+} \tag{14}$$

CVaR Prediction Model Based on Deep Learning

The continuous development of deep learning and the in-depth discussion of related theories have enabled the adoption of intelligent algorithms to solve some complex optimization problems (Guo et al., 2018). As a branch of machine learning, deep learning usually applies the structure of artificial neural network and corresponding intelligent algorithms to learn and characterize some difficult data.

CNN, as a special network model that combines deep learning and artificial neural network, is designed by people based on visual models. Its particularity is that the method of extracting relevant features of CNNs is different from other neural networks. The neurons are not fully connected and the weights between neurons are shared. CNNs can better reflect the mapping existing between input and output. After massive sample data are selected for training, the CNN will have the corresponding mapping ability. Therefore, there is no need to express a specific relationship between input and output.

The training of the CNN consists of two parts. First, the data signal propagates from a lower level to a higher level, which is the forward propagation part. The input signal x will go through the convolutional layer, the pooling layer, and the fully connected layer to finally get the output signal y. During forward propagation, the convolution kernel will perform operations on the extracted feature map of the previous convolutional layer, and the result obtained will generate the feature map of this layer through the action of the activation function. Assuming that there are q convolution kernels, there are also q corresponding feature maps, and the calculation method of the corresponding convolution layer is shown in Eq. (15).

$$a_{i,j}^{(p,q)} = f\left(\sum_{t=0}^{q_d-1}\sum_{x=0}^{q_h-1}\sum_{y=0}^{q_w-1}\omega_{\mathrm{x,y}}^{(\mathrm{q,t})} a_{i+x,j+y}^{(p-1,\mathrm{t})} + m^{(p,q)}\right) \tag{15}$$

where, the output of the i-th row and j-th column of the q-th feature map of the p-th convolutional layer is represented by a. The depth of the convolution kernel is represented by q_d, the height is represented by q_h, and the width is represented by q_w. The corresponding weight parameters are represented by ω. The offset term is represented by m. The activation function is represented by f.

The number of feature maps in the forward propagation process of the pooling layer won't change, but the size of the feature maps will change. The calculation method of the corresponding pooling layer is shown in Eq. (16).

$$b_{i,j}^{(p,q)} = f\left(\omega^{(q)}\sum_{x=0}^{s_h}\sum_{y=0}^{s_w} a_{i*s_w+x,j*s_h+y}^{(p-1,\mathrm{q})} + m^{(p,q)}\right) \tag{16}$$

where, the output of the i-th row and j-th column of the q-th feature map of the p-th pooling layer is represented by b. The height of the pooling window is represented by s_h, and the width is represented by s_w.

The output of the fully connected layer is processed by the activation function after the input weighted summation, and the corresponding calculation method is shown in Eq. (17).

$$c_{(i,j)} = f\left(\sum_{q=0}^{s-1}\sum_{x=0}^{s_h}\sum_{y=0}^{s_w}\omega_{\mathrm{x,y}}^{(\mathrm{j,q})} a_{x,y}^{(p-1,\mathrm{q})} + m^{(p,q)}\right) \tag{17}$$

where, the output of the j-th neuron of the i-th fully connected layer is represented by $c_{(i,j)}$. The number of input feature maps is represented by s, the height of input feature maps is represented by s_h, and the height of output feature maps is represented by s_w.

The back propagation of the CNN must first calculate the cost function. The form of the cost function is shown in Eq. (18).

$$D = \frac{1}{2g}\sum\nolimits_{h=1}^{g}\sum\nolimits_{l=1}^{k}\left(y_q^h - y_q^{*h}\right)^2 \tag{18}$$

where, the number of samples in the training set is represented by g. The number of classification categories is represented by k. The classification value of sample h is represented by Y^h_q. If the label of sample h happens to be k, Y^h_q is 1, otherwise Y^h_q is 0. The output value of the model relative to the k-th category of the sample h is represented by Y^{*h}_q.

To calculate the relative parameter gradient of the cost function, it is necessary to know the sensitivity of the q-th layer of the neural network, as shown in Eq. (19).

$$\lambda^p = \frac{\partial D}{\partial u^p} \tag{19}$$

where, λ^p represents the gradient of the cost function relative to the net activation, and the net activation is expressed by u^p.

The back propagation process will first pass through the fully connected layer, so the sensitivity of the fully connected layer will be calculated first. The calculation method is shown in Eq. (20).

$$\lambda^p = \left(\omega^{p+1}\right)^T \lambda^{p+1} \circ f'\left(u^p\right) \tag{20}$$

The in the equation represents the multiplication of the corresponding elements. The sensitivity of the output layer is shown in Eq. (21).

$$\lambda^P = \left(y - y^*\right) \cdot f'\left(u^P\right) \tag{21}$$

The partial derivative of the cost function with respect to the weight and bias term is shown in Eq. (22) and Eq. (23).

$$\frac{\partial D}{\partial w^p} = x^{p-1}\left(\lambda^p\right)^T \tag{22}$$

$$\frac{\partial D}{\partial m^p} = \frac{\partial D}{\partial \lambda^p}\frac{\partial \lambda^p}{\partial m^p} = \lambda^p \tag{23}$$

When the gradient of the pooling layer is calculated, it is also necessary to calculate the sensitivity of the pooling layer p, which also requires the sensitivity of the convolutional layer $p+1$. Since the convolution calculation is performed from the pooling layer p to the convolutional layer $p+1$, it is necessary to find the connection between the sensitivity of the current layer and the sensitivity of the next convolutional layer, that is, the relationship between the net activation of the convolutional layer *p+1* and the net activation of the pooling layer *p* needs to be obtained, which is shown in Eq. (24).

$$u_j^{p+1} = f^{'}\left(u_j^{p}\right) * w_j^{p+1} + m_j^{p+1} \tag{24}$$

where, the weight of the convolution kernel is represented by *w*, and its bias term is represented by *m*.

According to the relationship between the sensitivity of the convolutional layer and the sensitivity of the pooling layer, the sensitivity of the pooling layer is derived as shown in Eq. (25), where r180 indicates that the convolution kernel is rotated by 180 degrees.

$$\lambda_j^{p} = f^{'}\left(u_j^{p}\right) \circ \lambda_j^{p+1} * r180\left(w_j^{p+1}\right) \tag{25}$$

According to the sensitivity of the pooling layer, the calculation method of the bias term of the cost function relative to the pooling layer and the parameter gradient of the weight is shown in Eqs. (26) and (27).

$$\frac{\partial D}{\partial \omega_j^{p}} = \sum_{u,v}\left(u_j^{p} \circ down\left(x_j^{p-1}\right)\right)_{u,v} \tag{26}$$

$$\frac{\partial D}{\partial m_j^{p}} = \sum_{u,v}\left(\lambda_j^{p}\right)_{u,v} \tag{27}$$

When the sensitivity of the pooling layer is calculated to deduce the sensitivity of the convolutional layer, since the pooling layer will compress the corresponding data, the sensitivity map of the pooling layer will be smaller than the sensitivity map of the convolutional layer. The sensitivity map of the pooling layer is up-sampled to make it have the same size as the sensitivity map of the convolutional layer. Then, the sensitivity obtained from the up-sampling is multiplied term by term with the partial derivative of the activation function of the pooling layer to get the corresponding sensitivity, as shown in Eq. (28).

$$\lambda_j^{p} = up\left(\lambda_j^{p+1}\right) \circ f^{'}\left(u_j^{p}\right) \tag{28}$$

where, the up-sampling function is represented by *up*. After the sensitivity of the convolutional layer is acquired, the parameter gradient of the cost function relative to the convolutional kernel is obtained by using the chain derivative, as shown in Eq. (29).

$$\frac{\partial D}{\partial q_{x,y}^{p}} = \sum_{a}\sum_{b}\left(\lambda_{a,b}^{p}\lambda_{a+x-1,b+y-1}^{p-1}\right) \tag{29}$$

After the gradient of the cost function with respect to all parameters is acquired, the parameters of the neural network are updated, and the parameter update method is shown in Eq. (30) to Eq. (32).

$$\Delta q_{x,y}^{p} = -\eta \frac{\partial D}{\partial q_{x,y}^{p}} \tag{30}$$

$$\Delta m^{p} = -\eta \frac{\partial D}{\partial m^{p}} \tag{31}$$

$$\Delta \omega^{p} = -\eta \frac{\partial D}{\partial \omega^{p}} \tag{32}$$

The learning rate in Eqs. (30) to (32) is represented by η, which also represents the adjustment step of parameter update. Too small learning rate will result in slow model training speed, and too large learning rate will cause the system to be difficult to converge.

Through the establishment of a deep learning CNN model, the assumption of the distribution of financial asset returns and the estimation of variance in the traditional method are discarded. A time series prediction model and a regression prediction model are established for the loss sequence itself. First, the financial asset's rate of return is squared to get its loss sequence. Then, the loss sequence is lagged by one and two periods to obtain the time series with different lag periods, and a time series model based on the CNN is established. Finally, the loss sequence is regarded as the dependent variable, and the time sequence under the lag period is taken as the independent variable to establish the corresponding deep learning model, thereby obtaining the functional regression model based on the CNN.

PSO Algorithm for Computational Solution of Portfolio Model

Particle swarm algorithm is essentially an intelligent simulation algorithm. As a result of imitating the natural world, it utilizes particles to communicate and cooperate with each other to find the global optimal solution. This is also the essence of most intelligent cluster algorithms based on deep learning to seek optimal strategies.

Suppose that there is a population of birds in a certain space, and the birds in the population look for food in this space (Guo et al., 2019). No bird knows the location of the food, but it can judge the distance between its location and the location of the food. At the same time, it also knows the location closest to the food that other birds have been to and the best location in the entire population. Therefore, it can search for the location closest to the food it has been to or search for the current global best location to determine the location of the food. The corresponding particle swarm algorithm treats each bird as a particle, which is the solution in the search space (Jiang et al., 2017). The fitness of each particle is different, depending on the particle position and optimization function. The particles will travel towards the food position at a certain speed according to the best position they have been to and the current best position in the world. When iterating, the individual best position and the global best position will be updated, and the speed and direction of the particles will also be updated. The goal is to find the optimal solution, which is the location of the food in the bird flock.

The updating method of velocity in the traditional most basic PSO algorithm is shown in Eq. (33), and the updating method of position is shown in Eq. (34).

$$v^{t+1} = wv^{t} + c_1 r_1 \left(P^{t} - Z^{t}\right) + c_2 r_2 \left(P^{t} - Z^{t}\right) \tag{33}$$

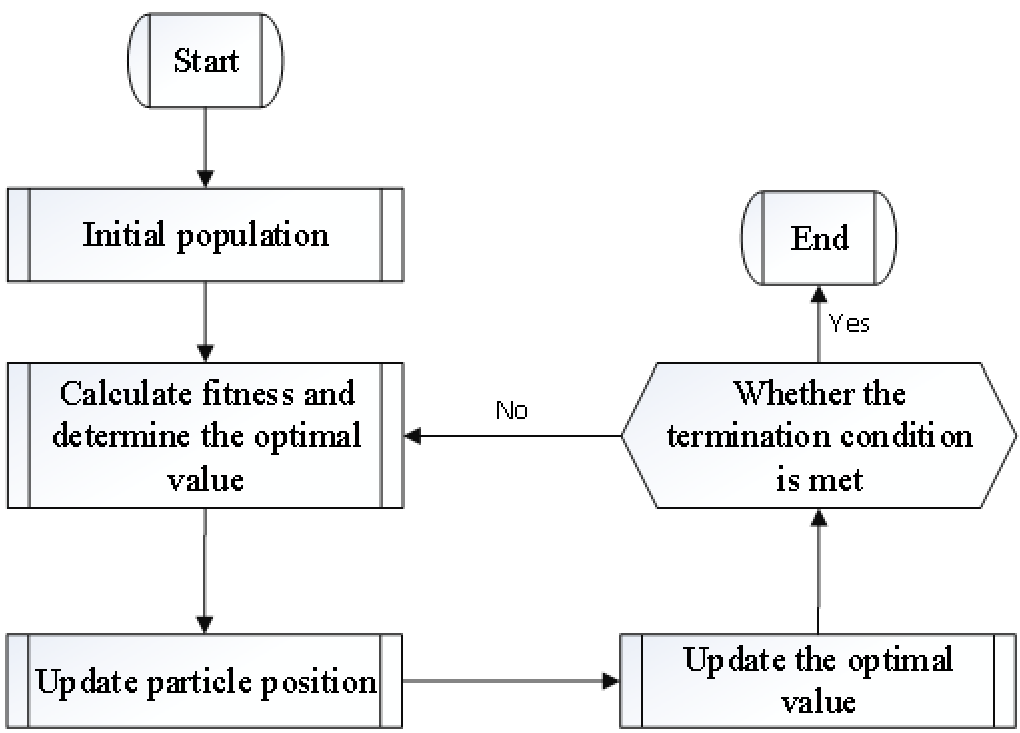
$$Z^{t+1} = Z^{t} + v^{t+1} \tag{34}$$

where, the number of iterations is represented by t, the inertia weight is expressed in terms of w, r_1 and r_2 are random numbers between 0 and 1, and the learning factors c_1 and c_2 are both positive values.

According to Eq. (33), the updating equation of particle velocity is divided into three parts. The first part is the velocity left over from the previous time due to inertia, also known as the memory term, and the magnitude of the inertial velocity is related to the characteristics of the particle itself; the second part is the learning of the best place they have been to, also known as individual cognitive term, and the speed of this part is affected by the learning factor c_1; the third part represents the influence of particles on the global optimal position, namely the recognition of global search, also known as the social cognition term, and the speed of this part is affected by the learning factor c_2. These three parts will act on the particle at the same time, to determine the velocity and direction of the particle after iteration and make it move towards the optimal solution.

The flow of traditional PSO algorithm based on deep learning is shown in Fig. 2.

Figure 2. Process of traditional particle swarm algorithm



According to Fig. 2, the traditional PSO algorithm first needs to initialize the original population, which requires all parameters to be initialized, including the area to search for the target, which must be bounded. Then there is the initialization of the learning factor, the iteration times of the algorithm should also be initialized, the precision of convergence also needs to be initialized to a constant, and the velocity of particles also needs to be defined by upper and lower limits. Of course, the key is initializing the velocity vector and position vector of each particle in the population.

Secondly, the corresponding fitness function needs to be set, and the fitness of each particle is calculated according to this fitness function. At the same time, the best location where each particle has been and the best fitness of the particle itself should be memorized. It is also necessary to retain the best position and the best fitness in the entire population. Then, the speed and position of the particles are updated iteratively according to the speed update equation and the position update equation of the particles in the population. The algorithm termination condition is that the number of iterations is completed or the searched optimal solution meets the accuracy requirements. The algorithm is terminated when either of these two conditions are met, otherwise it will continue to update iteratively.

The calculation of CVaR can be transformed into the calculation of the minimum value of a continuous function, which is not that bad, although the objective function is more complex. In order to improve the search efficiency of PSO algorithm, the PSO algorithm based on deep learning is enhanced and the problem that PSO algorithm is easy to fall into the local optimal solution is solved.

The inertia weight strategy is improved, and the weight change after the improvement is shown in Eq. (35).

$$w\left(t\right)=\begin{cases}\frac{t}{T}+0.7 & 0\le\frac{t}{T}\le 0.5\\ 1.2-\frac{t}{T} & 0.5\le\frac{t}{T}\le 1\end{cases} \tag{35}$$

where, the inertia weight of the degraded particles is directly cleared, and the optimal position of the objective function is the optimal solution in the whole world. In the particle swarm algorithm, the position of each particle is judged according to its fitness. In many cases, the fitness value of the particle swarm algorithm is a certain objective function, or the objective function is simplified as the fitness function. To maximize the return rate of the portfolio model and minimize the financial risk, CVaR can be directly used as the fitness function. There must be some constraints on the portfolio, the proportion of each asset must be no less than 0, and the sum must be equal to 1. However, the requirement of rate of return can't be directly limited in the process. Therefore, penalty function is usually used to improve the fitness function, which is used to reduce the fitness of the corresponding particles according to the penalty coefficient according to the magnitude of the violation of the constraint conditions. In this study, the fitness function is shown in Eq. (36) after adding penalty coefficient.

$$\min F=F\left(x,\eta\right)+e\cdot\max\left\{\rho-x^{T}R,0\right\} \tag{36}$$

where, the penalty coefficient is expressed by e, and the final particle position obtained by Eq. (35) is the optimal solution obtained by the PSO algorithm. The last item is the VaR value, and the previous one is the weight of each asset in the portfolio. Substituting these values into Eq. (36), the corresponding fitness can be obtained, which is the value of the objective function CVaR in this study (Liu & Chen, 2021).

RESULTS AND DISCUSSION

Test Results of Traditional Particle Swarm Algorithm And Enhanced Particle Swarm Algorithm

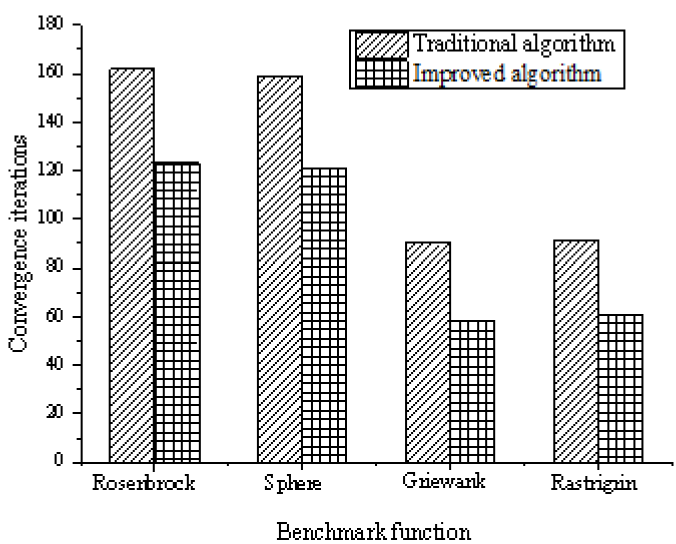
In this study, the performance testing was firstly conducted on the proposed PSO algorithm based on deep learning and the traditional particle swarm algorithm. The benchmark functions used for testing were currently four commonly used functions, namely, Griewank function, Rastrigrin function, Rosenbrock function, and Sphere function, among which the Griewank function and Rastrigrin function were multimodal functions, while the Rosenbrock function and the Sphere function are unimodal functions. The constraint intervals and the optimal values of these four benchmark functions are shown in Table 1.

Table 1. Constraint intervals of the four benchmark functions and their optimal solutions

Benchmark function	Bound interval	Optimal solution
Griewank function	[-600,600]	0
Rastrigrin function	[-5,5]	0
Rosenbrock function	[-10,10]	0
Sphere function	[-10,10]	0

In this study, the traditional particle swarm algorithm and the proposed PSO algorithm are initialized with the same parameters. The population size is set to 40; the particle dimension is 10; and the maximum number of iterations is set to 600. The traditional PSO algorithm and the PSO algorithm proposed in this study are respectively run for 15 times on the four test functions shown in Table 1, and the results of the 15 runs are counted. The simulation experiment in this study is realized by MATLAB system. The performance test results of the PSO algorithm proposed in this study based on deep learning and the traditional PSO algorithm are shown in Fig. 3.

Figure 3. Comparison of test results of traditional algorithms and optimized algorithms on four benchmark functions

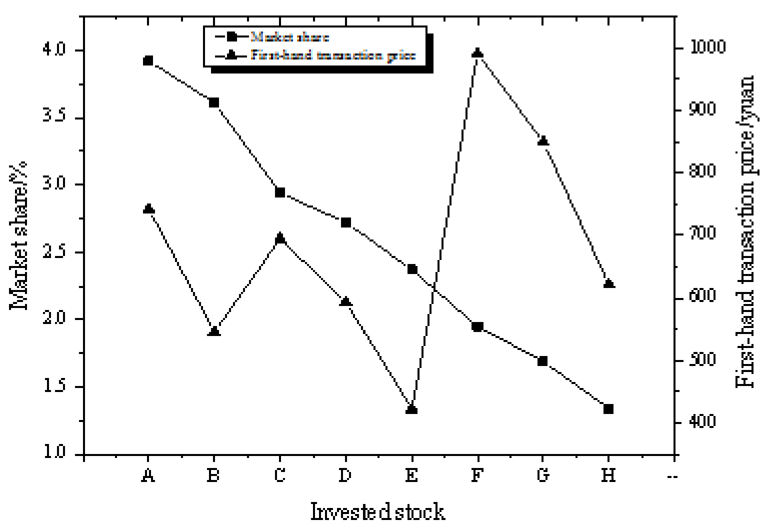


As shown in Figure 3, the two algorithms can obtain theoretically optimal solutions of the four test functions. However, according to the convergence rate, the traditional PSO algorithm can obtain the optimal solution for both Rosenbrock function and Sphere function at about 160 generations. For Griewank function and Rastrigrin function, convergence algorithm is almost around 90 generations. The PSO algorithm proposed in this study based on deep learning converges with Rosenbrock function and Sphere function in 120 generations, and the convergence speed is about 25% faster compared with the traditional PSO algorithm. For Griewank function and Rastrigrin function, it converges in the 60s, which is about 30% faster than the traditional PSO algorithm. The test results show that the PSO algorithm proposed in this study has a faster convergence rate and a better performance than the traditional PSO algorithm.

Analysis of the Solution Results of the Traditional Particle Swarm Algorithm to the Portfolio Model

According to the actual investment portfolio model, the performance of the particle swarm algorithm was measured in this work. Eight stocks were selected according to the principle of stock selection and the eight stocks were used to construct the portfolio model. The industries covered by stocks in the portfolio model selected in this study needed similar market capitalization proportions, and the smallest trading units of these stocks should have close trading prices, because constituent stocks had the smallest number of transactions in financial markets, and the number of trades put a certain limit on how many stocks can be traded. Therefore, the real income can not only prove the ratio of assets in the portfolio, but also the real income value of assets trading. If the price difference of the smallest trading unit is too large, the measurement of risk will be wrong and eventually lead investors to make wrong decisions about the investment portfolio. During the experimental verification in this study, it is necessary to ensure that the selected stocks will not be delisted or suspended from trading. According to these principles, the investment samples selected in this study are eight stocks of Bank of Communications (A), Min Sheng Bank (B), Industrial and Commercial Bank of China (C), Bank of Beijing (D), Agricultural Bank of China (E), CRRC corporation of China (F), Daqin Railway (G)

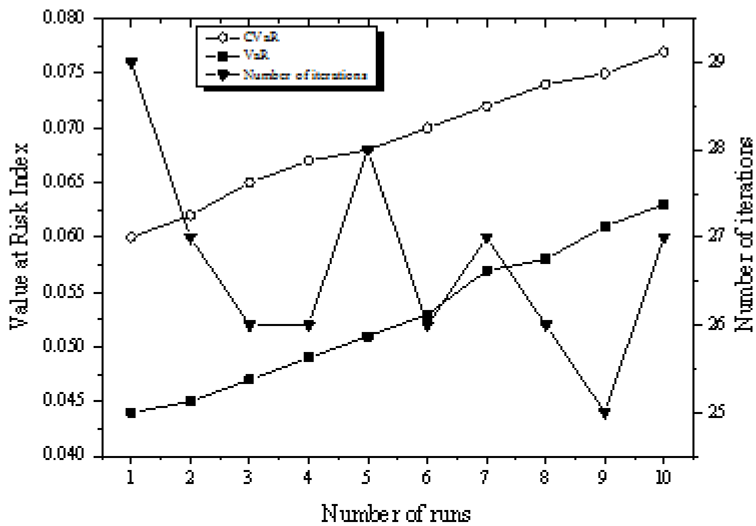
Figure 4. The market value of selected stocks and the first-hand transaction price



and Sinopec (H). The coverage of the stock industry and the proportion of market value are similar, and the market capitalization and average primary price are shown in Fig. 4.

According to the investment sample data proposed above, the traditional particle swarm algorithm is adopted to solve the investment portfolio CVaR prediction model based on CNN. The traditional PSO algorithm is run 10 times to get different portfolios according to the investors' input of different expected returns. The expected return rate entered by investors is higher and higher, and the risk measurement results obtained by the corresponding 10 algorithm runs are shown in Figure 5.

Figure 5. The risk evaluation results of the traditional particle swarm algorithm under different expected returns

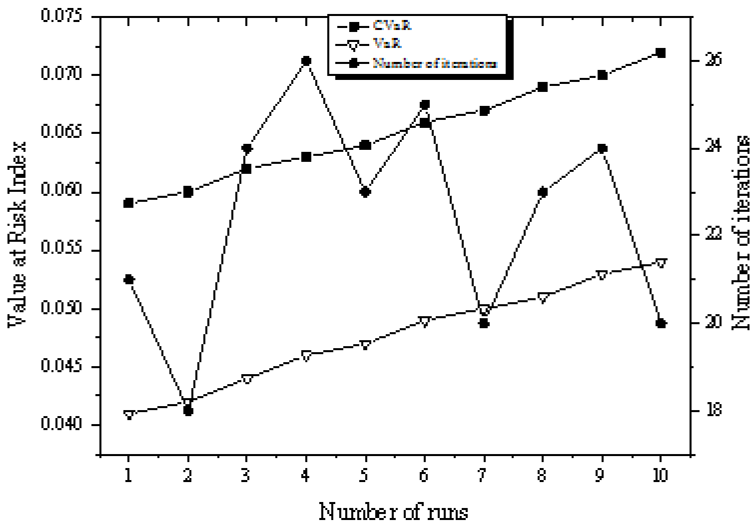


According to the investment sample data proposed above, the application of traditional PSO based on deep learning are shown in Figure 5. The higher the yield an investor expects, the greater the investment risk he faces. According to the establishment rules of the deep learning PSO algorithm and the portfolio model, the value of VaR is always smaller than the value of CVaR, as shown in Figure 5. When the expected rate is increasing, the traditional PSO algorithm can obtain the optimal solution of the portfolio model in a certain number of iterations, which also proves the feasibility of PSO based on deep learning in the risk assessment of financial investment that the portfolio can change the ratio between portfolios according to the financial risk, and it can also adjust the return rate of investors' expected assets to achieve more effective management of financial risks in the portfolio. The portfolio model of CVaR is solved. The traditional PSO algorithm is run 10 times to get different portfolios according to the investors' input of different expected returns. The expected return rate entered by investors is higher and higher, and the risk measurement results obtained by the corresponding 10 algorithm runs are shown in Figure 5.

Analysis of the Solution Result of the Improved Particle Swarm Algorithm to the Portfolio Model CVaR

According to the investment sample data proposed above, the PSO algorithm proposed was adopted to solve the investment portfolio CVaR prediction model based on CNN. The operation was the same as that of the traditional PSO algorithm. The investors input different expected return rates and run the PSO algorithm proposed in this study based on deep learning for 10 times respectively, which ended up with different portfolios. The corresponding investment risks were measured, as shown in Figure 6.

Figure 6. Risk evaluation results of PSO algorithm under different expected returns



According to Figure 6, among the risk evaluation results obtained by the PSO algorithm proposed in this study based on deep learning, the value of its VaR is also less than the value of CVaR, and the value of its VaR and CVaR also increase with the increase of investors' expected return rate, which further illustrates the fact that the higher the risk, the higher the risk. The PSO algorithm proposed in this study based on deep learning is the fastest, and it only cycles for 18 generations before the convergence algorithm can get the optimal solution, and this kind of optimization solution is already very fast, suggesting that the portfolio CVaR prediction model based on the CNN can use the improved PSO algorithm to obtain the optimal solution, and obtain the portfolio with relatively small risk loss, which is convenient for the risk management of the Internet financial market.

Comparison Result of Traditional Particle Swarm Algorithm and Improved Particle Swarm Algorithm

To compare the performance of the proposed PSO algorithm and the traditional particle swarm algorithm in practice, according to the selected investment samples, the PSO algorithm proposed, and the traditional particle swarm algorithm were adopted to optimize the CVaR prediction model of the CNN-based investment portfolio.

In this work, the PSO algorithm and the traditional particle swarm algorithm were adopted to run the CVaR prediction model of the portfolio based on the CNN 10 times. Then, the average value of the optimal solution of the portfolio model obtained by the algorithm was compared in Figure 7.

Figure 7. Comparison of the average value of the optimization algorithm and the traditional algorithm to find the optimal solution for the portfolio model

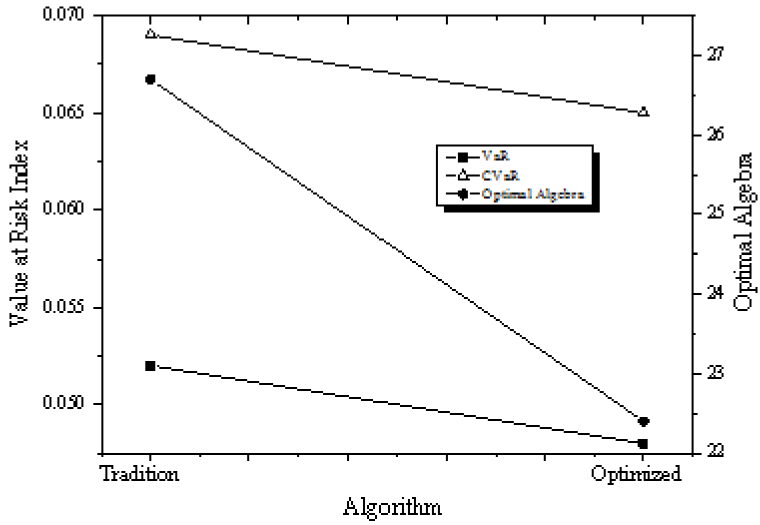


Figure 7 shows that the CVaR of the portfolio obtained by the proposed PSO algorithm is smaller than that of the traditional PSO algorithm. When adopted to obtain the optimized solution of the investment portfolio model, the PSO algorithm has obtained a smaller value-at-risk index relative to that of traditional particle swarm algorithm. It also indicates that the PSO algorithm proposed in this study based on deep learning can obtain a more optimized portfolio with lower investment risk while ensuring the expected rate of return. Moreover, the comparison of optimization cycle algebra between the two algorithms in Figure 8 shows that, the PSO algorithm based on deep learning is applied less cyclic algebra than the traditional PSO algorithm after finding the optimal solution. In summary, the CVaR prediction model of the portfolio based on the CNN can better manage the risk in the optimization of the portfolio in the real financial market.

CONCLUSION

In this study, to effectively manage the risks in the Internet financial market, the risks in the financial market stock portfolio is controlled based on deep learning methods in combination with PSO algorithms (Deng et al., 2021). First, a portfolio model was established based on CVaR. Then, a portfolio CVaR forecast model based on CNN was established, and the particle swarm algorithm was improved to optimize the solution. Moreover, four benchmark algorithms were employed to test its performance. Finally, the traditional PSO algorithm was compared with the PSO algorithm based on the actual financial market stock investment (Chen et al., 2020). The results prove that the improved PSO algorithm can solve the risks in the investment portfolio in a quicker and more effective manner. The CVaR prediction model of the CNN-based investment portfolio facilitates the risk management of the financial market. In short, this research provides an important reference for the adoption of deep learning methods and PSO algorithms in the Internet financial market investment portfolio risk management (Shen et al., 2019). However, there are still some deficiencies that other financial risks are neglected but introduces conditional value at risk to measure the risks in the portfolio. In the future, other risks in financial markets should be added for further research.

ACKNOWLEDGMENT

This work was supported by the Ministry of Education's Humanities and Social Sciences Research Outstanding Young and Middle-aged Ideological and Political Theory Teachers' Choice Funding Project "Looking for a "Living" History-Research on the Experiential Teaching Model of "The Outline of Modern Chinese History"" (No. 18JDSZK136) Research results.

REFERENCES

Al-Sulttani, A. O., Ahsan, A., Hanoon, A. N., Rahman, A., Daud, N. N. N., & Idrus, S. (2017). Hourly yield prediction of a double-slope solar still hybrid with rubber scrapers in low-latitude areas based on the particle swarm optimization technique. *Applied Energy*, *203*, 280–303. doi:10.1016/j.apenergy.2017.06.011

Almahdi, S., & Yang, S. Y. (2019). A constrained portfolio trading system using particle swarm algorithm and recurrent reinforcement learning. *Expert Systems with Applications*, *130*, 145–156. doi:10.1016/j.eswa.2019.04.013

Amirhosseini, B., & Hosseini, S. H. (2018). Scheduling charging of hybrid-electric vehicles according to supply and demand based on particle swarm optimization, imperialist competitive and teaching-learning algorithms. *Sustainable Cities and Society*, *43*, 339–349. doi:10.1016/j.scs.2018.09.002

Bai, B., Guo, Z., Zhou, C., Zhang, W., & Zhang, J. (2021). Application of adaptive reliability importance sampling-based extended domain PSO on single mode failure in reliability engineering. *Information Sciences*, *546*, 42–59. doi:10.1016/j.ins.2020.07.069

Chen, M., Liu, Q., Huang, S., & Dang, C. (2020). Environmental cost control system of manufacturing enterprises using artificial intelligence based on value chain of circular economy. *Enterprise Information Systems*, ●●●, 1–20. doi:10.1080/17517575.2020.1856422

Chiang, H. S., Sangaiah, A. K., Chen, M. Y., & Liu, J. Y. (2020). A novel artificial bee colony optimization algorithm with SVM for bio-inspired software-defined networking. *International Journal of Parallel Programming*, *48*(2), 310–328. doi:10.1007/s10766-018-0594-6

Das, S. P., & Padhy, S. (2018). A novel hybrid model using teaching–learning-based optimization and a support vector machine for commodity futures index forecasting. *International Journal of Machine Learning and Cybernetics*, *9*(1), 97–111. doi:10.1007/s13042-015-0359-0

Deng, X., Guo, X., Wu, Y. J., & Chen, M. (2021). Perceived environmental dynamism promotes entrepreneurial team member's innovation: Explanations based on the uncertainty reduction theory. *International Journal of Environmental Research and Public Health*, *18*(4), 2033. doi:10.3390/ijerph18042033 PMID:33669732

Faia, R., Pinto, T., Vale, Z., & Corchado, J. M. (2018). Strategic particle swarm inertia selection for electricity markets participation portfolio optimization. *Applied Artificial Intelligence*, *32*(7-8), 745–767. doi:10.1080/08839514.2018.1506971

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669. doi:10.1016/j.ejor.2017.11.054

Gao, W., & Su, C. (2020). Analysis of earnings forecast of blockchain financial products based on particle swarm optimization. *Journal of Computational and Applied Mathematics*, *372*, 112724. doi:10.1016/j.cam.2020.112724

Ghasemiyeh, R., Moghdani, R., & Sana, S. S. (2017). A hybrid artificial neural network with metaheuristic algorithms for predicting stock price. *Cybernetics and Systems*, *48*(4), 365–392. doi:10.1080/01969722.2017.1285162

Ghimire, S., Deo, R. C., Raj, N., & Mi, J. (2019). Wavelet-based 3-phase hybrid SVR model trained with satellite-derived predictors, particle swarm optimization and maximum overlap discrete wavelet transform for solar radiation prediction. *Renewable & Sustainable Energy Reviews*, *113*, 109247. doi:10.1016/j.rser.2019.109247

Gong, X. L., Liu, X. H., Xiong, X., & Zhuang, X. T. (2019). Forecasting stock volatility process using improved least square support vector machine approach. *Soft Computing*, *23*(22), 11867–11881. doi:10.1007/s00500-018-03743-0

Guo, Y., Han, S., Shen, C., Li, Y., Yin, X., & Bai, Y. (2018). An adaptive SVR for high-frequency stock price forecasting. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 11397–11404. doi:10.1109/ACCESS.2018.2806180

Guo, Y., He, J., Xu, L., & Liu, W. (2019). A novel multi-objective particle swarm optimization for comprehensible credit scoring. *Soft Computing*, *23*(18), 9009–9023. doi:10.1007/s00500-018-3509-y

Han, Y., Zhang, F., Huang, L., Peng, K., & Wang, X. (2021). Does industrial upgrading promote eco-efficiency?— A panel space estimation based on Chinese evidence. *Energy Policy*, *154*, 112286. doi:10.1016/j.enpol.2021.112286

He, H., Zhang, W., & Zhang, S. (2018). A novel ensemble method for credit scoring: Adaption of different imbalance ratios. *Expert Systems with Applications*, *98*, 105–117. doi:10.1016/j.eswa.2018.01.012

Jiang, S., Chin, K. S., Wang, L., Qu, G., & Tsui, K. L. (2017). Modified genetic algorithm-based feature selection combined with pre-trained deep neural network for demand forecasting in outpatient department. *Expert Systems with Applications*, *82*, 216–230. doi:10.1016/j.eswa.2017.04.017

Korovkinas, K., Danėnas, P., & Garšva, G. (2020). Support vector machine parameter tuning based on particle swarm optimization metaheuristic. *Nonlinear Analysis (Vilnius)*, *25*(2), 266–281. doi:10.15388/namc.2020.25.16517

Latchoumi, T. P., Ezhilarasi, T. P., & Balamurugan, K. (2019). Bio-inspired weighed quantum particle swarm optimization and smooth support vector machine ensembles for identification of abnormalities in medical data. *SN Applied Sciences*, *1*(10), 1–10. doi:10.1007/s42452-019-1179-8

Liu, Y., & Chen, M. (2021). Applying text similarity algorithm to analyze the triangular citation behavior of scientists. *Applied Soft Computing*, *107*, 107362. doi:10.1016/j.asoc.2021.107362

Ma, T., Wang, F., Wang, J., Yao, Y., & Chen, X. (2017). A combined model based on seasonal autoregressive integrated moving average and modified particle swarm optimization algorithm for electrical load forecasting. *Journal of Intelligent & Fuzzy Systems*, *32*(5), 3447–3459. doi:10.3233/JIFS-169283

Maiyar, L. M., Cho, S., Tiwari, M. K., Thoben, K. D., & Kiritsis, D. (2019). Optimising online review inspired product attribute classification using the self-learning particle swarm-based Bayesian learning approach. *International Journal of Production Research*, *57*(10), 3099–3120. doi:10.1080/00207543.2018.1535724

Niu, Z., Zhang, B., Li, D., Ji, D., Liu, Y., Feng, Y., Zhou, T., Zhang, Y., & Fan, Y. (2021). A mechanical reliability study of 3-dB waveguide hybrid couplers in submillimeter and terahertz bands. *Frontiers of Information Technology & Electronic Engineering*, *22*(8), 1104–1113. doi:10.1631/FITEE.2000229

Shen, C. W., Chen, M., & Wang, C. C. (2019). Analyzing the trend of O2O commerce by bilingual text mining on social media. *Computers in Human Behavior*, *101*, 474–483. doi:10.1016/j.chb.2018.09.031

Shi, K., Tang, Y., Liu, X., & Zhong, S. (2017). Secondary delay-partition approach on robust performance analysis for uncertain time-varying Lurie nonlinear control system. *Optimal Control Applications & Methods*, *38*(6), 1208–1226. doi:10.1002/oca.2326

Song, Y., Zhang, F., & Liu, C. (2020). The risk of block chain financial market based on particle swarm optimization. *Journal of Computational and Applied Mathematics*, *370*, 112667. doi:10.1016/j.cam.2019.112667

Tang, B. Q., Han, J., Guo, G. F., Chen, Y., & Zhang, S. (2019). Building material prices forecasting based on least square support vector machine and improved particle swarm optimization. *Architectural Engineering and Design Management*, *15*(3), 196–212. doi:10.1080/17452007.2018.1556577

Wang, B., Ma, F., Ge, L., Ma, H., Wang, H., & Mohamed, M. A. (2020). Icing-EdgeNet: A pruning lightweight edge intelligent method of discriminative driving channel for ice thickness of transmission lines. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–12. doi:10.1109/TIM.2020.3044719

Xiang, D., Zhang, Y., & Worthington, A. C. (2021). Determinants of the use of fintech finance among chinese small and medium-sized enterprises. *IEEE Transactions on Engineering Management*, *68*(6), 1590–1604. doi:10.1109/TEM.2020.2989136

Yi, H. (2021). Secure social internet of things based on post-quantum blockchain. *IEEE Transactions on Network Science and Engineering*, 1. Advance online publication. doi:10.1109/TNSE.2021.3095192

Zhang, H. (2020). Optimization of risk control in financial markets based on particle swarm optimization algorithm. *Journal of Computational and Applied Mathematics*, *368*, 112530. doi:10.1016/j.cam.2019.112530

Zhou, H., Sun, G., Fu, S., Liu, J., Zhou, X., & Zhou, J. (2019). A big data mining approach of PSO-based BP neural network for financial risk management with IoT. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 154035–154043. doi:10.1109/ACCESS.2019.2948949

HsienChen Liu is the corresponding author of this paper. E-mail: 52214600002@stu.ecnu.edu.cn.