



Forecasting stock volatility process using improved least square support vector machine approach

Xiao-Li Gong¹ · Xi-Hua Liu¹ · Xiong Xiong^{2,3} · Xin-Tian Zhuang⁴

Published online: 5 January 2019

© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

Considering that the stock returns distribution displays leptokurtosis as well as left-skewed properties, and the returns volatility process exhibits heteroscedasticity as well as clustering effects, the asymmetric GARCH-type models with non-Gaussian distributions (AGARCH-nG) are employed to describe the volatility process. In addition, the AGARCH-nG models are hybridized with artificial neural network (ANN) technique for forecasting stock returns volatility. Since the least square support vector machine (LS-SVM) technique displays strong forecast ability, we present an improved particle swarm optimization (IPSO) algorithm to optimize the parameters of LS-SVM technique in the process of stock returns volatility prediction. Then, we compare the forecasting performances of individual AGARCH-nG models, the hybrid AGARCH-nG-ANN methods and the data mining-based LS-SVM-IPSO method using stock markets data. The empirical results verify the effectiveness and superiority of the proposed method, which demonstrates that the LS-SVM-IPSO approach outperforms the AGARCH-type models with non-Gaussian distributions and those integrating with the artificial neural network methods.

Keywords Stock volatility forecasting · Leptokurtosis distribution · Artificial neural network · Least square support vector machine · Particle swarm optimization algorithm

1 Introduction

The stock market is an extremely complex nonlinear dynamic system, in which the ability of forecasting stock returns volatility with great precision is of significant importance for investors and regulators. Volatility modeling and prediction of stock index returns are important subjects in financial risk management including derivative pricing, risk measurement and multi-period portfolio

selection. Since it can proxy the risk conditions in financial markets, it has attracted a great deal of attentions. Taking into account the stylized facts of financial phenomena, the GARCH-type family models are extensively used for capturing the clustering properties in financial asset returns volatility prediction (Bentes 2015a, b; Abounoori et al. 2016; Lv and Shan 2013). However, the GARCH family models belong to parametric models which require prior structure assumptions of financial data. Considerable researchers have found that financial time series data does not follow normal distribution assumption. On the contrary, it exhibits leptokurtosis and heavy tailed properties in addition to excessive skewness and kurtosis. Hence, non-Gaussian distributional assumptions need to be imposed to the GARCH-type family processes to capture the left-skewed distributional characteristics of financial returns. Besides, the GARCH family processes cannot describe the nonlinear complex dependence relationships among financial asset variables, which may lead to unsatisfactory forecasting consequences. To overcome the drawbacks of conventional financial forecasting models and to enhance

Communicated by V. Loia.

✉ Xi-Hua Liu
xihua-liu@163.com

¹ School of Economics, Qingdao University, Qingdao 266061, China

² College of Management and Economics, Tianjin University, Tianjin 300072, China

³ China Center for Social Computing and Analytics, Tianjin 300072, China

⁴ School of Business Administration, Northeastern University, Shenyang 110169, China

the volatility prediction ability, it calls for more advanced techniques to identify these complex patterns.

With rapid developments of artificial intelligent technologies, the artificial neural network techniques and machine learning techniques have been attracting more and more attentions in recent years and have been widely applied in finance fields (Hajizadeh et al. 2012; Bildirici and Ersin 2009; Kristjanpoller et al. 2014), especially for financial asset returns and volatility predictions (Lahmiri 2016a, b). The integration of artificial intelligent technologies with conventional GARCH-nG models can effectively improve the volatility forecasting performances of stock indices and has become hot research topics (Tseng et al. 2009). Furthermore, intelligent optimization algorithms comprising genetic algorithm, particle swarm optimization algorithm and so on have been utilized to improve the estimation accuracy of intelligent algorithm parameters.

However, it may not always appropriate to choose forecasted series of GARCH-type families as inputs to artificial neural network for further volatility predictions. Because the financial asset variables typically display nonlinear and non-Gaussian behaviors, it requires two aspects of extensions. One research extension is the abnormal distribution assumptions for modeling of returns time series distribution, and the other research direction is the assumption-free techniques to identify the evolutionary dynamic pattern for the volatility process. Hence, our research will also focus on these two aspects to compare the forecasting ability of AGARCH-nG-ANN models with the improved support vector machine method based on modified optimization algorithm.

Support vector machine is a rather powerful machine learning method based on the statistical learning theory, which takes the structural risk minimization as the principle, and effectively suppresses the over-learning phenomenon, thus displaying good generalization capabilities. The training of the model is transformed into solving a quadratic programming problem to ensure the global optimality and to solve the local minimum problem in the neural network. The LS-SVM (Suykens et al. 2002) is the support vector machine that takes quadratic loss functions as empirical risks. It replaces inequality constraints with equality constraint and transforms the training of the model into linear solutions to the equations, which simplifies the calculation process and shortens the training time with more deterministic training consequences so that it is suitable for online applications. Therefore, this study investigates the predictive power of LS-SVM based on improved particle swarm optimization algorithm in stock index volatility forecasting, and the outcomes are contrasted with those of the hybrid approaches.

Since the previous works about volatility forecasting mainly involve artificial neural network hybridized with GARCH model without non-Gaussian innovations or merely involve the support vector machine technique without parameter optimization. In this paper, we construct the LS-SVM method optimized by the modified PSO algorithm to enhance the returns volatility forecasting. In addition, we incorporate the non-Gaussian innovations to the previous AGARCH-ANN models. Then, we compare the volatility forecasting performances of the proposed model with other two extended models using stock market data from wind database, namely the asymmetric GARCH models with non-Gaussian distributions hybridized with artificial neural network and the individual parametric asymmetric GARCH models. The forecasting accuracy results provide suggestions for the stock market investors.

2 Literature review

Because of the high peakness and excessive kurtosis, financial time series data are generally not normally distributed. Several works have examined the influences of different choices of distribution forms. Generally, the generalized error distribution assumption is made for innovations since its capability of characterizing excessive kurtosis, and the Student-*t* distribution assumption is made for innovations to flexibly characterize the fat tails of financial time series. Zhu and Galbraith (2010) proposed the asymmetric Student-*t* (AST) distribution which allows separate parameters to control skewness and thickness of the tails. Through exploiting the asymmetrical and fat-tailed models, researchers have found evidence supporting the usefulness of such extensions in describing the complex pattern of financial data (Alberg et al. 2008).

There has been growing interests in modeling nonlinear models, and nonparametric models for volatilities prediction have gone through major developments (Kristjanpoller and Minutolo 2016). Researchers have made efforts to take advantage of the artificial intelligence techniques for applications in finance and economics (Cheng and Wei 2009; Ramyar and Kianfa 2017). Previous studies have found that the hybrid model integrating asymmetric GARCH models and the artificial neural network techniques display much more closeness to the actual volatility process (Kristjanpoller et al. 2014), which indicate that the GARCH-type ANN models provide better prediction performances than pure GARCH-type models. Especially when extending the asymmetric GARCH family models with artificial neural network approaches, it can effectively enhance the forecasting ability. However, when using extracted technique indicator as inputs to ANN with back-propagation neural network, Lahmiri (2017) found that it

produced higher accurate estimation consequences than the hybrid EGARCH-ANN approaches. Currently, the ANN is one of the most widely used nonlinear forecasting models that deal with the complex correlation pattern among variables, especially applications in stock price predictions (Qiu et al. 2016; Zahedi and Rounaghi 2015; Göçken et al. 2016). Ince et al. (2017) investigated the predictive accuracy of exchange rate with ANN and the monetary model. Choudhary and Haider (2012) assessed the power of several ANN combination models for inflation forecasting and found that these may serve as credible tools. Dhamija and Bhalla (2010) employed data during financial meltdown period to demonstrate the usefulness of GARCH models and artificial neural network technique in determining the long-run nonlinearity of sample data. Monfared and Enke (2015) presented an adaptive neural network filter so as to predict the error in GARCH models and then applied it to predict the GARCH process.

However, the ANN technique has certain troublesome problems such as difficulties in determining the number of hidden layer nodes in neural network, the existence of excessive learning phenomenon and the local minimum problem in the training process. Fortunately, the support vector machine method is developed from statistical learning theory, which has obvious advantages in the prediction of small samples, nonlinear and high-dimensional problems. Researchers have extensively applied SVM in the fields of economics and finance, such as making economic forecasts and predicting bankruptcy (Zhao et al. 2017). With the further study of SVM methods, various improvement methods have been proposed and applied (Rojo-Álvarez et al. 2014). It is illustrated that the LS-SVM method has good robustness and is suitable for large-scale computing. Zhao et al. (2016) stated that it is more advantageous to use LS-SVM as the nonlinear prediction model, but it also needs modifications to improve functionalities. Ismail et al. (2011) proposed the integration of self-organizing maps and LS-SVM technique for time series forecasting. The PSO algorithm has been widely used in recent years (Abualigah and Khader 2017; Abualigah et al. 2018). In order to improve the classification accuracy, Liu and Zhou (2015) introduced PSO algorithm into the LS-SVM model for novel data classification approach. With reference to the above ideas, this paper presents an improved PSO algorithm to optimize the LS-SVM parameters so as to enhance the accuracy of predictions. Furthermore, it is noted that few studies have attempted to comprehensively compare parametric approaches and nonparametric approaches in stock volatility forecasting, our study will contribute to the selection of methods in financial volatility prediction techniques.

The motivation of this study is to enhance the returns volatility forecasting of previous methods and investigate whether the improvements can be achieved by the LS-SVM technique optimized by IPSO algorithm. In order to compare the different performance properties of LS-SVM-IPSO and hybrid AGARCH-nG-ANN as well as the AGARCH family models, three kinds of models are constructed to conduct the stock index volatility forecasting.

3 GARCH-type models with non-Gaussian distributions

In this paper, three volatility forecast techniques including the traditional GARCH-type processes, the hybrid artificial neural network using GARCH-nG prediction outcomes as inputs and the IPSO-modified LS-SVM method are compared to predict stock index historical volatility. In this section, we will focus on the extensions of GARCH-type family models. A critical feature of the substantial progress has been the incorporation of fatter tails and asymmetric effects into the variance process, which will better improve the fitting of the tail decay rates since declines in financial returns may involve more extreme returns movements.

4 Asymmetric GARCH models

For the sake of capturing the clustering property and heteroscedasticity effects, the time changing volatility process σ_t can be expressed in GARCH (m, n) models (Bollerslev 1986) as follows, which depends on p order past conditional variance as well as q order past squared innovations.

$$\begin{aligned}\sigma_t^2 &= \omega_1 + \sum_{i=1}^p \alpha_{1i} \varepsilon_{t-i}^2 + \sum_{i=1}^q \beta_{1j} \sigma_{j-i}^2 \\ \varepsilon_t &= \sigma_t Z_t\end{aligned}\quad (1)$$

where Z_t is i.i.d. random variables with zero mean and unit variance, ε_t represents uncorrelated series with zero mean and variance σ_t , $\alpha_{1i} > 0$, and $\beta_{1j} > 0$ as well as the following constraints are satisfied to ensure stability.

$$\sum_{i=1}^p \alpha_{1i} + \sum_{i=1}^q \beta_{1j} < 1 \quad (2)$$

In order to better capture the significant asymmetry in asset returns caused by asymmetric effects of different shocks, the EGARCH and GJR-GARCH models are most frequently employed. The last two versions can account for the leverage effects of returns on conditional variance, which means that the large returns decline may cause larger return volatility than the increases with the same amplitude.

Since the EGARCH model (Nelson 1991) describes the variance as the asymmetric function of disturbance error ε_t , it can be used to account for the asymmetric effects in volatility, capturing the asymmetric behaviors of returns, which can be expressed in the following form:

$$\log(\sigma_t) = \omega_2 + \sum_{i=1}^p \alpha_{2i} \left[\frac{|\varepsilon_{t-i}|}{\sqrt{\sigma_{t-i}}} - \sqrt{\frac{2}{\pi}} \right] + \gamma_{2i} \frac{\varepsilon_{t-1}}{\sqrt{\sigma_{t-1}}} + \sum_{j=1}^q \beta_{2j} \log(\sigma_{t-j}) \tag{3}$$

where ω_2 , α_{2i} and β_{2j} denote the model parameters with no restrictions, respectively.

Another model that can capture the asymmetric characteristics of returns is the GJR-GARCH model, which allows the conditional variance to react differently to the negative and positive innovations. It can be expressed as follows:

$$\sigma_t^2 = \omega_3 + \sum_{i=1}^p \beta_{3i} \sigma_{t-i}^2 + \sum_{i=1}^q \alpha_{3i} \varepsilon_{t-i}^2 + \sum_{i=1}^q \gamma_{3i} \varepsilon_{t-i}^2 \Theta \tag{4}$$

where $\Theta = 1$ if $\varepsilon_{t-i} < 0$, and $\Theta = 0$ if $\varepsilon_{t-i} \geq 0$, $\alpha_{3i} \geq 0$, $\beta_{1j} \geq 0$ and $\alpha_{3i} + \gamma_{3i} \geq 0$. Besides, the following constraints need to be satisfied.

$$\sum_{i=1}^p \alpha_{3i} + \sum_{i=1}^q \beta_{3j} + \frac{1}{2} \sum_{i=1}^q \gamma_{3j} < 1 \tag{5}$$

5 Non-Gaussian innovation distributions

Considering that the financial returns series distribution typically displays leptokurtosis and heavy tailed characteristics in addition to excessive kurtosis, the non-Gaussian distributional assumptions are often imposed to the GARCH-type volatility models (Chuang et al. 2007), such as the Student- t distribution, the generalized error distribution (GED) and the generalized asymmetric Student- t distribution (Zhu and Galbraith 2011). The different hypotheses of the error distributions in GARCH-type models are to capture different statistical features. The asymmetric GARCH models allow various features of returns distributions by specifying the distributions of innovations.

For the stock returns series y_t with mean μ and variance σ , the Student- t density distribution function has the following expression.

$$f(y, \mu, \sigma, \nu) = \frac{\Gamma(\nu + 1/2)}{\sigma^2 \sqrt{\pi(\nu - 2)} \Gamma(\nu + 1/2)} \left(1 + \frac{(y - \mu)^2}{\sigma^2(\nu - 2)} \right)^{-\frac{(\nu+1)}{2}} \tag{6}$$

where Γ represents the gamma function, and the degree of freedom parameter that controls the thickness satisfies the condition of $\nu > 2$.

For the generalized error distribution assumption of returns distribution, it has the following density function.

$$f(y, \mu, \sigma, \theta) = \frac{\theta}{2\sigma\Gamma(1/\theta)} \exp\left(-\frac{|y_t - \mu|^\theta}{\sigma}\right) \tag{7}$$

where θ represents the control parameter.

The generalized asymmetric Student- t distribution allows separate parameter to control the thickness and skewness of tails, which is important for quantities relying on tail features. The general form of the AST density in rescaled version can be expressed as follows:

$$f_{AST}(y) = \begin{cases} \frac{1}{\sigma} \left[1 + \frac{1}{\nu_1} \left(\frac{y - \mu}{2\alpha_4 \sigma K(\nu_1)} \right)^2 \right]^{-(\nu_1+1)/2}, & \text{if } y \leq \mu \\ \frac{1}{\sigma} \left[1 + \frac{1}{\nu_2} \left(\frac{y - \mu}{2(1 - \alpha_4) \sigma K(\nu_2)} \right)^2 \right]^{-(\nu_2+1)/2}, & \text{if } y > \mu \end{cases} \tag{8}$$

where $K(\nu) = \Gamma((\nu + 1)/2) / [\sqrt{\pi\nu} \Gamma(\nu/2)]$, Γ represents the gamma function and $\alpha \in (0, 1)$, $\nu_1 > 0$ and $\nu_2 > 0$.

5.1 GARCH-nG parameter estimation method

For the selection of the optimal parameter lag for the GARCH-nG models, the penalized model order selection criteria containing the Bayesian information criterion (BIC) and the Akaike's information criterion (AIC) are employed.

Traditionally, the maximum likelihood estimation method is employed for parameter estimations of GARCH-type variance equation. And the estimations are obtained through numerical maximization of each of the corresponding likelihood functions. Let the parameter vector of the likelihood function be Ψ , and then the general likelihood function form can be expressed as follows.

$$l(\Psi; y) = \sum_{t=1}^T \left\{ \log \delta - \log \sigma_t + \log f_Y \left(\tau + \delta \frac{r_t - \mu}{\sigma_t} \right) \right\} \tag{9}$$

where τ denotes the mean value of the density function f_Y , and δ denotes the standard deviation of the density function f_Y .

6 Least square support vector machine with improved particle swarm optimization

There often exist nonlinear patterns in financial time series that cannot be characterized by parametric models. Hence, the approximation of the real relationships in financial data

calls for more advanced methods. In this study, we mainly focus on artificial intelligence-based ANN methods and the machine learning-based LS-SVM method. Specifically, the ANN is hybridized with the asymmetric GARCH models with non-Gaussian innovations. And we propose the least square support vector machine technique with improved particle swarm optimization, in which the multi-region adaptive PSO algorithm is employed for parameter estimations.

6.1 Artificial neural network

The artificial neural network has unique advantages in dealing with stock price forecasting, which is a multi-factor, uncertain and nonlinear time series forecasting problem. Volatility changes display nonlinear features, and the transaction information contains a large number of decisions inherently in stock returns volatility changes. Through the study of historical data, neural network can search the rules and patterns of parameters autonomously from the complex financial time series data.

The ANN that depends on no hypotheses tries to mimic the structure of biological neural network, where a set of neurons are connected in layers. It has the advantages of adaption on the basis of the properties extracted from the research problem. The typical ANN that is organized hierarchically is generically composed of one input layer that has x variables, one hidden layer and the output layer that has forecasted variable y calculated as follows.

$$y_i = f\left(\sum_{j=1}^p x_j w_{ij} + \eta\right) \quad (10)$$

where w_{ij} denotes the connection weights between neuron j and neuron i , η is the bias, and f denotes the activation function affecting the output amplitude. The input layer of ANN corresponds to the input variables, and the hidden layer is employed for capturing nonlinear relationships.

Typically, the sigmoid function is chosen as well as the Levenberg–Marquadt algorithm being employed for network training. And the connecting weights are adjusted according to the following rule.

$$\Delta w_i = -(M_i^T M_i + cI)^{-1} M_i e_i \quad (11)$$

in which M is the Jacobian matrix, I represents the identity matrix, and c denotes the adaptive learning parameter.

The commonly used ANN techniques comprise the BP neural network (BP-NN), the wavelet neural network and the RBF neural network, among which the BP-NN technique is chosen in this work since it is the most extensively employed technique. Theoretically, the BP-NN is capable of approximating any complicated nonlinear functions, displaying strong generalization abilities and self-learning features. The back-propagation neural network takes inputs

from the previous layer and then sends outputs to the next layer. The BP neural network technique is introduced into the stock volatility analysis and evaluation to understand the internal mechanism of the volatility dynamics so as to effectively predict the volatility changes.

6.2 Least square support vector machine

However, the ANN also has drawbacks of over-fitting problems. Additionally, the BP neural network has the disadvantages of easily falling into local solutions and slow convergence speed in handling with high-dimensional data (Dai et al. 2014). Whereas the support vector machine is a typical machine learning algorithm with complete mathematic theories and excellent learning abilities that have become a hot research topic in machine learning fields and have been successfully applied in many fields. The SVM takes the training error as the constraint condition of the optimization problem and regards the minimization of the confidence range as the optimization goal. Namely, it is a learning method based on the structural risk minimization criterion, whose promotion ability is obviously superior to the neural network. It is assumed that there are two types of sample points in the two-dimensional space, and the data samples are mapped into high-dimensional space through nonlinear functions. Subsequently, a pair of parallel hyperplanes are constructed to separate the training samples and to maximize the distance between each point in the sample set and the hyper-plane.

The SVM training solves a constrained quadratic programming problem, and the number of constraints is equal to the sample size. Therefore, when the sample size is fairly large, it may take long time for training. To improve the training efficiency of SVM, the least square support vector machine is applied based on SVM. Compared with the SVM, the LS-SVM replaces slack variables by square of training error and uses equality constraints instead of inequality constraints. The training process only requires solving a linear system of equations, which avoids the time-consuming QP problem. Furthermore, it displays simple calculation and fast convergence speed with high precision that has been widely applied in the fields of nonlinear process modeling. Since this paper focuses on nonlinear process modeling based on the LS-SVM regression, its basic principle is primarily described below.

Suppose that the training sample set is $\{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$, where $x_i \in \mathbb{R}^N$ stands for the input of the i th sample and $y_i \in \mathbb{R}$ stands for the output of the i th sample, $i = 1, 2, \dots, l$. For the nonlinear system, the nonlinear function is assumed to be expressed in the following form.

$$f(x) = w^T \cdot \varphi(x) + b \quad (12)$$

where $\varphi(\cdot)$ denotes the kernel space mapping function that maps the input data of the original space to the high-dimensional feature space. The slack variable of the SVM optimization problem is replaced by the square error term and the inequality constraint is varied to equality constraint. Then, the optimization problem of the LS-SVM regression is expressed as follows.

$$\begin{aligned} \min_{w,b,e} J(w, e) &= \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^l e_i^2 \\ \text{s.t. } y_i &= w^T \varphi(x_i) + b + e_i, \quad i = 1, 2, \dots, l \end{aligned} \tag{13}$$

where the weight vector $w \in R^N$; the error variable $e_i \in R$; b represents the deviation amount; C stands for the regularization parameter.

$$L(w, b, e; \alpha) = J(w, e) - \sum_{i=1}^l \alpha_i (w^T \varphi(x_i) + b + e_i - y_i) \tag{14}$$

According to the above optimization function, the Lagrange function can be defined as follows.

where the Lagrange multiplier (i.e., support vector) $\alpha_i \in R$. According to the KKT conditions, the following are available.

$$\begin{cases} \frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^l \alpha_i \varphi(x_i) = 0 \\ \frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^l \alpha_i = 0 \\ \frac{\partial L}{\partial e_i} = 0 \rightarrow \alpha_i = C e_i \\ \frac{\partial L}{\partial \alpha_i} = 0 \rightarrow w^T \varphi(x_i) + b + e_i - y_i = 0 \end{cases} \tag{15}$$

Eliminate variables w, e_i , the matrix equation can be derived in the following.

$$\begin{bmatrix} 0 & 1 & 1 & \dots & 1 \\ 1 & K(x_1, x_1) + \frac{1}{2C} & K(x_1, x_1) & \dots & K(x_1, x_n) \\ 1 & K(x_2, x_1) & K(x_2, x_2) + \frac{1}{2C} & \dots & K(x_2, x_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & K(x_n, x_1) & K(x_n, x_2) & \dots & K(x_n, x_n) + \frac{1}{2C} \end{bmatrix} \tag{16}$$

$$\begin{bmatrix} b \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} 0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{17}$$

where $K(x, x_i) = \langle \varphi(x), \varphi(x_i) \rangle$ in Eq. (16) stands for the kernel function that represents the inner product of the high-dimensional feature space. According to the functional theory, as long as satisfying the Mercer conditions,

the function can be used as kernel functions, which commonly contains the linear functions, the perceptron functions and the radial basis functions. In view of better performances of radial basis functions, the radial basis function $K(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / (2\sigma^2))$ is chosen as the kernel function of LS-SVM. Finally, the decision function is determined as shown in Eq. (18):

$$f(x) = \sum_{i=1}^n \alpha_i K(x, x_i) + b \tag{18}$$

The selection of kernel function causes great influences on the generalization ability of the system. Hence, the selection of kernel function constitutes the core issue of the support vector machine theories. Unfortunately, there has been no effective ways to choose kernel functions so far. In practice, there are several commonly used kernel functions in the following forms:

- (1) Polynomial kernel function: $K(x_i, x) = [(x_i \cdot x) + 1]^q$, where q stands for the polynomial parameter;
- (2) Radial basis function: $K(x_i, x) = \exp(-\|x_i - x\|^2 / \sigma^2)$, where σ denotes the radial basis function parameter;
- (3) Sigmoid kernel function: $K(x_i, x) = \tanh(v(x_i \cdot x) + c)$, where $v > 0, c < 0$.

What can be seen is that the number of intermediate nodes in LS-SVM is finally transformed into solving a convex optimization problem, which is theoretically optimal. As shown in the LS-SVM structure in Fig. 1, $x = (x^1, x^2, \dots, x^m)$ denotes the input vector, m represents the dimension of the input vector, n is the number of support vectors, and the number of nodes in the middle layer is n with $n \neq m$.

6.3 Improved particle swarm optimization algorithm

In general, the commonly used methods of parameter estimation are as follows: (1) the least square estimate of linear method. The advantage of this method is the simple

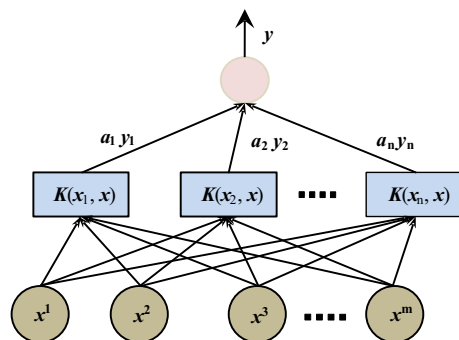


Fig. 1 The structure of LS-SVM

calculation process, in which only large equations need to be solved. On the contrary, the disadvantage is that the regression equation has large residuals and relatively low prediction accuracy. (2) the Quasi-Newton method for solving unconstrained optimization problem. However, during the parameter optimization process, this method becomes easy to fall into local extremum, so that the obtained model may deviate from reality. At the same time, this method also requires partial derivative of the optimized function, which is difficult to be satisfied in practical problems. Concerning how to find a fast and effective algorithm to estimate the parameters is an important research issue. Fang et al (2010) pointed out that the determination of parameters belonged to the nonlinear combinatorial optimization problem, and most of the optimal solutions of the objective function to the parameters were pathological problems with no analytic solutions. Therefore, it requires the introduction of advanced and flexible optimization algorithm such as the intelligent optimization algorithm. However, the direct use of particle swarm optimization algorithm for parameter estimation has the disadvantage of large estimation errors. In this section, some improvements are proposed and the improved algorithm is employed to estimate the parameters.

The basic PSO algorithm searches for the optimal solution by cooperation and information sharing among individual groups in every iteration, estimating the objective function of the particle and determining the best position $pbest$ of each particle at time t and the best position of the group $gbest$, and then iteratively update the particle velocity and position. As the particles in the PSO move toward the best position in their own history and gather around the best position in the cluster, the rapid convergence effects of the particle population are formed, which will also result in falling into the local optimal solution and the early convergence phenomenon. Moreover, the usage of real financial market data to estimate the nonlinear optimal model parameters can readily lead to the local optimal solution, resulting in inaccurate outcomes. Therefore, it is of significant importance to improve the computational efficiency of the PSO algorithm.

The basic particle swarm algorithm assumes that each particle flies in the n -dimensional space, $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ represents the current position of particle i , $V_i = (v_{i1}, v_{i2}, \dots, v_{in})$ represents the current flying speed of particle i , $p_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{in})$ denotes the optimal location for the individual particle i . The particles are dynamically adjusted according to individual and group flight experiences, with the velocity v_{ij} and the position x_{ij} upgrade equation satisfying Eq. (19)

$$\begin{aligned} v_{ij}(t+1) &= v_{ij}(t) + c_1 r_1 (pbest_{ij}(t) - x_{ij}(t)) \\ &\quad + c_2 r_2 (gbest_j(t) - x_{ij}(t)) \\ x_{ij}(t+1) &= x_{ij}(t) + v_{ij}(t+1) \end{aligned} \quad (19)$$

where c_1, c_2 stand for the learning factors, c_1 can adjust the flying distance of the particle to its best position, c_2 can adjust the particle's optimal flight step to the crowds, r_1, r_2 stand for the random numbers in $[0,1]$, $gbest_j(t)$ denotes the j th-dimensional component of the optimal position of the particle swarm at the t th iteration. The first term in the first formula represents the inertia term, and the second term represents the cognition term, which means learning from its own history, also known as individual cognition. The third expression of collective information indicates that it is based on group information, with the adjustment reflecting coordination between particles. The so-called sociality is essentially the enhancement of the probability of the implementation of behaviors when an individual sees other individuals strengthening certain behaviors.

In order to improve the particle swarm initialization accuracy and to reduce the post-particle search time, Yang and Lee (2012) proposed a multi-basin PSO algorithm to keep the population of particles exploring in multiple regions so as to avoid premature, preventing falling into the local optimum. In order to increase the particle diversity, Pehlivanoglu (2013) proposed a multi-frequency vibration PSO algorithm with parameter variation. Based on these methods and taking account of the idea of mutation, a multi-region adaptive PSO algorithm with parameter and population adaptive mutation is proposed. The multi-region search of the initial particles is carried out to improve the initialization effects, the adjustment probability is adaptively determined according to the convergence of the algorithm, and the chaos traversal properties are utilized to avoid falling into the local optimum. The improved particle swarm optimization algorithm includes the following operations: the population initialization with multi-region local search; the adaptive mutation of particle swarm parameter; the adaptive mutation of population global region.

6.3.1 Population initialization with multi-region local search

The linear local search method is utilized to find the local minimum from the randomly generated initial parameter set Θ_i . The iterative formula of partial search is $\Theta_{i+1} = \Theta_i + l_i d_i$, $d_i = -M(i)^{-1} \nabla f(\Theta_i)$, where l_i denotes the phase step, d_i denotes the gradient direction, $M(i)$ represents the non-singular square matrix, which is an approximation of the Hessian matrix satisfying positive definiteness, and $d_i^T \nabla f(\Theta_i) = -\nabla f(\Theta_i)^T M(i)^{-1} \nabla f(\Theta_i) < 0$.

The gradient direction d_i is updated at each iteration, and the phase step l_i satisfies the Wolfe condition, that is, $f(\Theta_i + l_i d_i) \leq f(\Theta_i) + z_1 l_i \nabla f(\Theta_i)^T d_i$, $\nabla f(\Theta_i + l_i d_i)^T d_i \geq z_2 + l_i d_i)^T d_i \geq z_2 \nabla f(\Theta_i)^T d_i$, where $z_1, z_2 \in [0, 1]$.

The individual region converged to the local extremum is defined as $W(\Theta_n) = \{\Theta \in R^n : \lim_{i \rightarrow \infty} \Theta_i = \Theta^n\}$. Based on the local search method with global convergence, the whole search space R^n is divided into several regions $W(\Theta_n)$, in which each region contains the optimal value of convergence of the initial particle swarm. And the initial particle swarm in the same region converges to the same objective function value. Then, the parameter set $S_1 = (\Theta_1^1, \Theta_2^1, \dots, \Theta_n^1)$ initialized after the local search is superior to the original parameter set $S_0 = (\Theta_1^0, \Theta_2^0, \dots, \Theta_n^0)$.

6.3.2 Parameter adaptive mutation

The particles show strong convergence and decreasing rates in following the optimal particles. In order to increase the diversity of particles and to prevent it from falling into the local optimum, adaptive crossover and mutation operators are introduced according to the distance between each particle and the optimal particle $L = \sqrt{\sum_{i=1}^D (x_{1i} - x_{2i})^2}$, adaptively determining the algorithm adjustment probability under convergence. The particles in the population are taken out sequentially to determine whether the spatial distance between the extracted particle and the backup multi-region initialized optimal particle is less than the threshold $\Delta\tau = (1 - t/T)^p \times (ub - lb)$. If it is less than the threshold, the crossover operation is performed to enhance the search of the middle region of the particle, where ub and lb are the upper and lower limits of the problem, p represents the adjustment parameter.

The initial particle size of the algorithm is diversified. At this time, the larger threshold value is suitable for population adjustment. As the iterations proceed, the population tends to aggregate and a small threshold needs to be set to allow more particles to cross and mutate. The cross-operation formula can be expressed as $bx_1 = x_1e + x_2(1 - e)$, $bx_2 = x_1(1 - e) + x_2e$, where bx_1, bx_2 denote new particles generated by crossover, and e stands for the random number sequence in $(0, 1)$. After the crossover operation, the new particle's fitness value is calculated. If the fitness value becomes better, the old particle is replaced by the new one. In contrast, if the fitness value becomes worse, then the mutation operation is introduced to enhance the fine search around the particle. The mutation operation can be expressed as $cx_1 = x + (1 - t/T)^q(ub - x)$, $cx_2 = x - (1 - t/T)^q(x - lb)$, where cx_1, cx_2 are new particles generated by mutation, q denotes the weight of variation. After the crossover and mutation operations, the global

optimal particle is moved relatively a little distance in the upper and lower limits to get a new set of particles and then recalculates the fitness value of the particles, as well as selecting the particles with better fitness value instead of the optimal particles.

6.3.3 Population global adaptive variation

For the sake of avoiding the algorithm getting into the local optimum, chaotic variables are introduced. By utilizing the chaotic ergodic properties, it leads the particle swarm to the optimal solution. According to the particle convergence trend and the fitness value differences in the population, mutation probability is set for each particle. Define the degree of convergence $O_t = 1/[1 + u \sqrt{(1/n)(\sum_{i=1}^n (fit_i^t - fit_{mean}^t)^2)}]$, where u is the adjustment factor, fit_i denotes the fitness value of the i th particle at the t th iteration, and fit_{mean} represents the average fitness value of all the particles after the t th iteration. When the degree of population aggregation increases, the differences of fitness values became smaller and the convergence degree O_t becomes larger. Set the mutation probability $P_{mu} = \sin((\pi/2) \cdot (f_{gbest}^t/f_i^t) \cdot O_t)$. When $rand < P_{mu}$ is met, the particles are mutated and the local extremum can jump out. Chaos mutation is applied to the particles that meet the mutation conditions, with the formula of chaos variation expressed as $x_{ijm} = x_{ijmin}^t + \rho_{ij}^{t+1}(x_{ijmax}^t - x_{ijmin}^t)$, $x_{ij}^{new} = 0.5(x_{ij} - x_{ijm})$, where ρ_{ij}^{t+1} is the chaotic variable traversed in $(0, 1)$, x_{ijm} denotes the value of the i th particle after the variation in the j th dimension, x_{ijmin}^t and x_{ijmax}^t , respectively, represent the minimum and maximum values experienced in the t th iteration and x_{ij}^{new} is the new particle generated after the mutation.

6.3.4 Multi-region adaptive PSO algorithm flow design

Step 1 Initialize the population randomly, update the iterative formula Θ_{i+1} , and adjust the particle population size and velocity V_i on the multi-region parameter space set Θ_n to obtain the number of particles backed up by the multi-region local search.

Step 2 Update particle velocity v_{ij} and position x_{ij} according to formula (18) and record the global optimal particle $gbest$ and the historical optimal particle $pbest$.

Step 3 Perform cross-mutation operation on the particles in the population according to the conditions.

Step 3.1 Determine whether the distance L between the particles taken out in sequence and the global optimum particle meets the threshold condition $\Delta\tau$. If not, move to the next particle, repeat step 3.1, otherwise go to the next step.

Step 3.2 Cross-operate on the particles satisfying the threshold condition $\Delta\tau$ to generate the new particle bx_i . If the fitness value gets better, replace the particle and move on to the next particle. Repeat from step 3.1, otherwise go to step 3.3.

Step 3.3 Perform mutation operation to generate new particles cx_i , calculate the new particle fitness value and compare it with the original particle, then replace the particle with poor fitness values, and turn to the next particle, repeat from step 3.1, update the global optimal particle $gbest$ and the particle historically optimum $pbest$.

Step 4 The mutation probability P_{mu} is calculated according to the convergence degree Q_t to determine whether the particle satisfies the chaotic mutation condition. If the particle satisfies the condition, the chaotic mutation x_{ijm} is applied for adaptive mutation, with the particle fitness values calculated. Then, the global optimal particle $gbest$ and the particle history optimum $pbest$ are updated.

Step 5 Determine whether the algorithm meets the termination criteria. If satisfied, then terminate the algorithm, output $gbest$ and $pbest$; if not satisfied, then $t = t + 1$, go to step 2.

7 Empirical results

7.1 Data description

In this section, the S&P 500 index (SP500) and the Chinese Shanghai and Shenzhen 300 composite index (HS300) are investigated. The study dataset includes the actual observation values from January 4, 2010 to December 29, 2017, among which the former 80% of the observation values are utilized for training and the last 20% observation values are utilized for forecast evaluations. The corresponding data are collected from the wind database. The basic statistical features of stock index returns are computed for both stock index series containing the volatility, skewness and kurtosis as well as the heteroscedasticity test and normality test. Figure 2 displays the SP500 and HS300 stock indices and the corresponding returns. Both the Ljung-Box $Q^2(10)$ statistics and the ARCH effects test for the squared returns indicate that both the returns indices exhibit strong heteroscedasticity effects. Additionally, the Jarque-Bera tests of both series reject the normal distribution, and the kurtosis values suggest that both the returns series have higher peakness compared to the Gaussian distribution.

There exist certain explanatory factors that influence the asset returns in financial markets significantly. Primarily, eight technical indicators are extracted from historical volatility series. The technical indicators analysis is based on statistical principles, employing a large number of historical data with addition, subtraction, multiplication and division of statistics and calculation methods to create a series of mathematical formula index system. The

Fig. 2 The stock price index and returns of HS300 and SP500

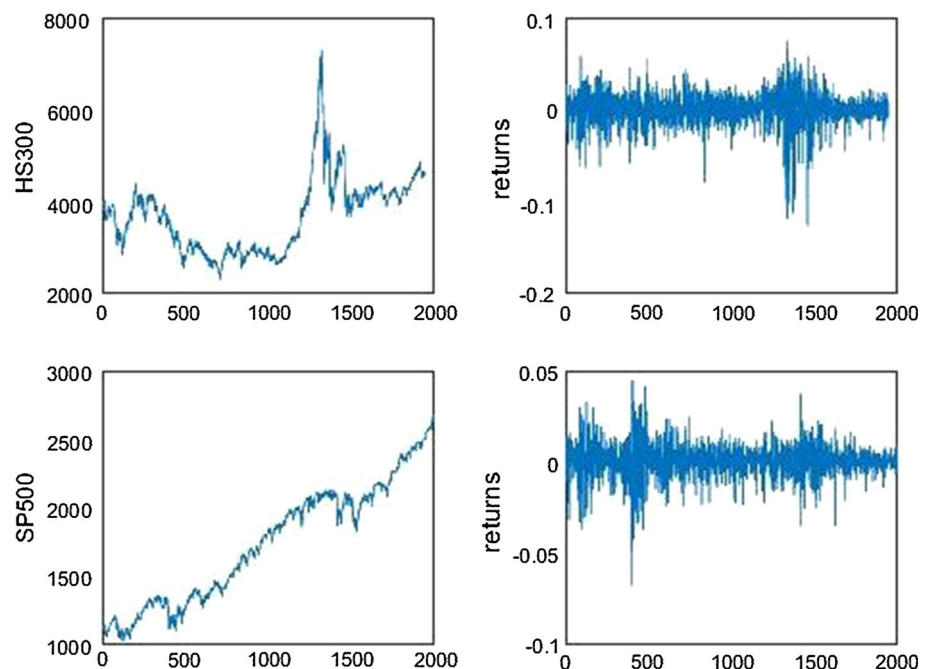


Table 1 Description of technical indicator

Technical indicator	Meaning	Technical indicator	Meaning
Moving average	The average stock price of a few days, which can observe the trend of the stock price	Trade volume index	Cumulation of the daily total stock market volume. When the closing price is higher than the previous one, the total volume is positive; or otherwise
Random index	Include the highest price and the lowest price in the calculation cycle, taking into account the random amplitude in the stock price volatility	Popular trading index	Use the highest point, the lowest point and the differences between open market and the ratio in a certain period to reflect the stock market strength, trading momentum.
Smoothing the difference with the moving average	Using the aggregation and separation function of fast moving average and slow moving average to conduct double smoothing operation to determine the time of buying and selling signals	Buyers willingness index	Reflect the strength of the trend between yesterday's closing price of the stock index and today's highest index and the lowest index, thus reflecting the stock index will
Relative strength Index	Compare the average closing up and the average closing down in a certain period to analyze the market's intentions and strength	Deviation rate	The relationship between the price index and the moving average ratio. The stock price deviation from the moving average level in order to determine the trading behavior of investors

combination of historical data and technical indicators for stock forecasting will make the results more effective. And Table 1 provides the descriptions of the technical analysis indicators.

7.2 Performance indicator

In order to compare the prediction performances by different approaches, the following indicators containing the mean absolute error (MAE), the mean of squared error (MSE) and the root mean square error (RMSE) are chosen to evaluate the error accuracy.

$$MAE = 1/n \sum_{i=1}^n |y_a - y_f| \tag{20}$$

$$MSE = 1/n \sum_{i=1}^n (y_a - y_f)^2 \tag{21}$$

$$RMSE = \left(1/n \sum_{i=1}^n (y_a - y_f)^2 \right)^{1/2} \tag{22}$$

where y_a denotes the actual values, y_f denotes the model forecast values, and n represents the observation number.

7.3 Experimental results and discussion

In the GARCH-nG method, the fitness of GARCH-nG, EGARCH-nG and the GJR-nG models are initially evaluated. According to the AIC and BIC criteria, the best parameter lag for GARCH-nG combined with various (p,q) orders ranging from (1,1) to (10,10) is calculated. And the best model specification to fit the distributional characteristics of the data is found based on a set of criteria. We report the corresponding estimation parameters of asymmetric GARCH-non-Gaussian models of SP 500 index and HS300 index in Tables 2 and 3. Furthermore, the volatility forecasting consequences using EGARCH parametric method with Student- t distribution, GED distribution and AST distribution are presented in Fig. 3, where the SP500 index volatility and HS300 index volatility are taken as examples.

In the second hybrid method, the obtained volatility estimations from GARCH-nG models complemented with technical indicators are fed to the neural network. The GARCH-nG models are hybridized with the ANN method and the realized volatility is regarded as the output target of the network. Namely, the hybrid method takes as inputs the

Table 2 Parameter estimation of EGARCH-non-Gaussian models of S&P500

Model	EGARCH-nG (Student- t , GED, AST) model								
	w_2	α_2	β_2	γ_2	ν	θ	α_4	ν_1	ν_2
SP500	0.0009 (0.0007)	0.2227 (0.0319)	0.6319 (0.9578)	0.0329 (0.0874)	6.0743 (0.7953)	2.9364 (0.0817)	0.4390 (0.0182)	6.5508 (0.8517)	15.1733 (5.9442)
HS300	0.0002 (0.0030)	0.0708 (0.0187)	0.9136 (0.0292)	0.0457 (0.0631)	6.4681 (0.8053)	2.6727 (0.0670)	0.4557 (0.0177)	6.6023 (0.8283)	15.2631 (6.0215)

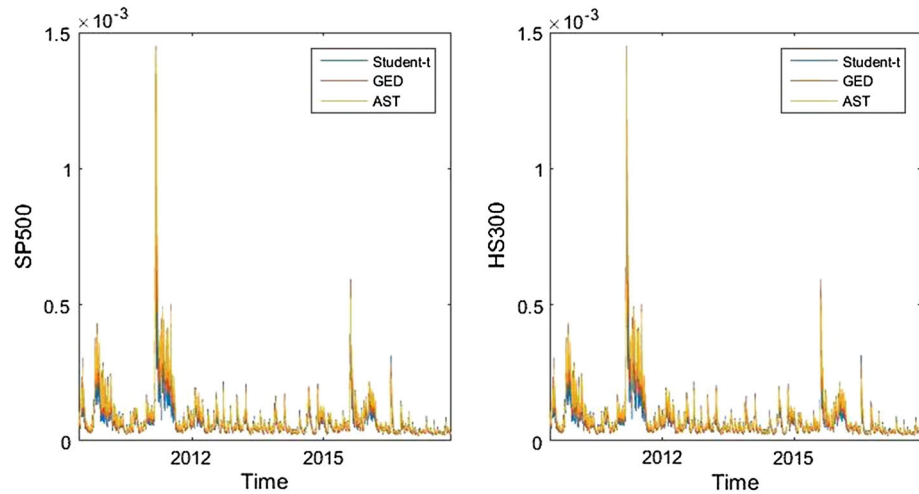
The value in parenthesis denotes the standard deviation error

Table 3 Parameter estimation of GJR-GARCH-non-Gaussian models of HS300

Model	GJR-GARCH-nG (Student- <i>t</i> , GED, AST) model								
Parameter	w_3	α_3	β_3	γ_3	ν	θ	α_4	ν_1	ν_2
SP500	0.0004 (0.0002)	0.1573 (0.0197)	0.8192 (0.0202)	0.0285 (0.0174)	2.7389 (0.3902)	2.8541 (0.0769)	0.5216 (0.0102)	6.4904 (0.8112)	16.2104 (6.5971)
HS300	-0.0004 (0.0004)	0.3912 (0.0081)	0.3683 (0.0012)	0.0316 (0.0224)	3.0117 (0.4128)	2.5217 (0.0548)	0.5668 (0.0094)	6.2206 (0.7945)	16.3872 (6.2450)

The value in parenthesis denotes the standard deviation error

Fig. 3 The volatility forecasting using EGARCH with different innovation distributions



simulated volatility as well as the specified explanatory variables for training the network. This hybrid method is capable of keeping the fine properties of GARCH-nG models while enhancing with the ANN technique. During the ANN running stage, the sample dataset is divided into the training period and the prediction period, in which the training data are utilized to determine model specifications, and the forecast data are reserved for evaluation. And repeated training is carried out in the hidden layer to determine appropriate numbers of neurons. When the error between the desired output and the actual output is less than the specified value or if the termination criterion is

satisfied, the network training is completed with the weights and bias being saved. The network with back-propagation is trained employing the Levenberg–Marquardt method, and the input variables are normalized between values of -1 and 1 .

In the third method, the proposed IPSO algorithm is applied to the LS-SVM for optimizing parameters. And the evolutions of fitness function with respect to generations of IPSO algorithm and PSO algorithm are compared in Fig. 4. What can obviously see from the plot is that the IPSO algorithm converges faster along with lower values than that of PSO algorithm. Since it uses the technical indicators

Fig. 4 The fitness functions of IPSO algorithm (left panel) and PSO algorithm (right panel)

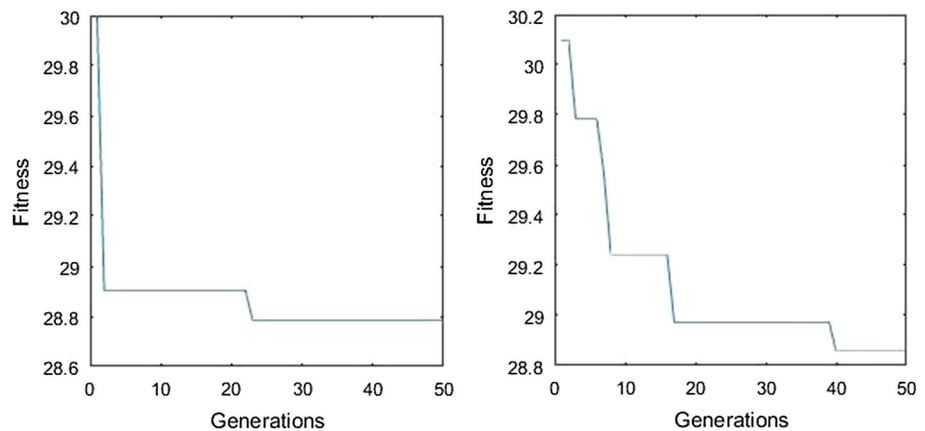


Table 4 The stock index volatility forecasting results of S&P 500

Model type	MAE	MSE	RMSE
GARCH type			
GARCH-normal	0.00726	0.0000827	0.00617
GARCH-GED	0.00715	0.0000746	0.00628
GARCH-Student- <i>t</i>	0.00712	0.0000806	0.00594
GARCH-AST	0.00710	0.0000795	0.00587
EGARCH-normal	0.0689	0.0000755	0.00600
EGARCH-GED	0.0651	0.0000713	0.00605
EGARCH-Student- <i>t</i>	0.0621	0.0000700	0.00588
EGARCH-AST	0.0607	0.0000685	0.00532
GJR-GARCH-normal	0.0703	0.0000788	0.00624
GJR-GARCH-GED	0.0679	0.0000739	0.00624
GJR-GARCH-Student- <i>t</i>	0.0646	0.0000721	0.00599
GJR-GARCH-AST	0.0617	0.0000694	0.00548
AGARCH-nG-ANN			
EGARCH-normal-ANN	0.00436	0.0000374	0.00499
EGARCH-GED-ANN	0.00416	0.0000369	0.00426
EGARCH-Student- <i>t</i> -ANN	0.00309	0.0000351	0.00415
EGARCH-AST-ANN	0.00257	0.0000288	0.00374
ML			
SVM-PSO	0.00184	0.0000068	0.00417
LS-SVM-IPSO	0.00083	0.0000029	0.00328

Table 5 The stock index volatility forecasting results of HS300

Model type	MAE	MSE	RMSE
GARCH type			
GARCH-normal	0.00837	0.0000854	0.00622
GARCH-GED	0.00797	0.0000803	0.00637
GARCH-Student- <i>t</i>	0.00743	0.0000766	0.00581
GARCH-AST	0.00713	0.0000789	0.00576
EGARCH-normal	0.0546	0.0000714	0.00613
EGARCH-GED	0.0522	0.0000698	0.00622
EGARCH-Student- <i>t</i>	0.0507	0.0000638	0.00576
EGARCH-AST	0.0487	0.0000602	0.00519
GJR-GARCH-normal	0.0714	0.0000740	0.00651
GJR-GARCH-GED	0.0695	0.0000736	0.00638
GJR-GARCH-Student- <i>t</i>	0.0651	0.0000732	0.00608
GJR-GARCH-AST	0.0622	0.0000697	0.00582
AGARCH-nG-ANN			
EGARCH-normal-ANN	0.00330	0.0000271	0.00485
EGARCH-GED-ANN	0.00388	0.0000246	0.00413
EGARCH-Student- <i>t</i> -ANN	0.00262	0.0000219	0.00406
EGARCH-AST-ANN	0.00248	0.0000274	0.00385
ML			
SVM-PSO	0.00147	0.0000059	0.00408
LS-SVM-IPSO	0.00051	0.0000017	0.00311

affecting the stock index returns volatility, it is expected that the model can better capture the impacts of market variables.

Tables 4 and 5 report the evaluation accuracy results according to the measurement metrics. The lower the measurement indicators are, the higher prediction accuracy of the model is. What can be seen from the results is that the AGARCH process models with asymmetric Student-*t* innovation distribution display slightly lower prediction errors than that with normal distribution for both the two stock indices. This finding suggests that the S&P 500 stock index and HS300 index exhibit fat-tailed properties, which require taking account of the non-Gaussian distributional assumptions to model the volatility dynamic process. Similarly, the asymmetric GARCH-type models provide lower measurement errors in case of modeling the stock index volatility, which suggests that it is appropriate to consider the asymmetric effects of volatility modeling. The consequences provide evidence for the usefulness of non-Gaussian distributions in fitting financial assets.

In addition, it can be concluded that the hybrid AGARCH-ANN models that utilize the outcomes of AGARCH family models as inputs perform better than the AGARCH models on the whole. And the hybrid AGARCH-ANN model has been proven robust for

different specifications and performs better than the GARCH-nG models in predicting the stock index volatility. The obtained consequences support the findings of previous works that the combination of asymmetric GARCH models with neural network can improve the prediction ability of the individual GARCH process models in forecasting the stock index volatility.

However, it is not always an efficient approach when compared to the modified machine learning methods. As indicated by the performance measurements, the LS-SVM-IPSO technique outperforms other methods by providing the lowest forecasting errors in forecasting the volatility of S&P 500 index and HS 300 index. Moreover, in contrast to the parametric method, the least square methods do not need to regard the type of the innovation distributions, showing flexibility implementation capability.

The relative errors of the predictions of the training set and the test set in the LS-SVM-IPSO model are less than those of the hybrid ANN models. The above comparison results show that the LS-SVM-IPSO model has more advantages than the hybrid ANN models. Furthermore, Figs. 5, 6 plot the predicted values employing the proposed method with the actual observation values. It is observed that the proposed LS-SVM-IPSO approach presents a powerful and efficient tool with very small deviations in

Fig. 5 The volatility forecasting and estimation errors of S&P500 index

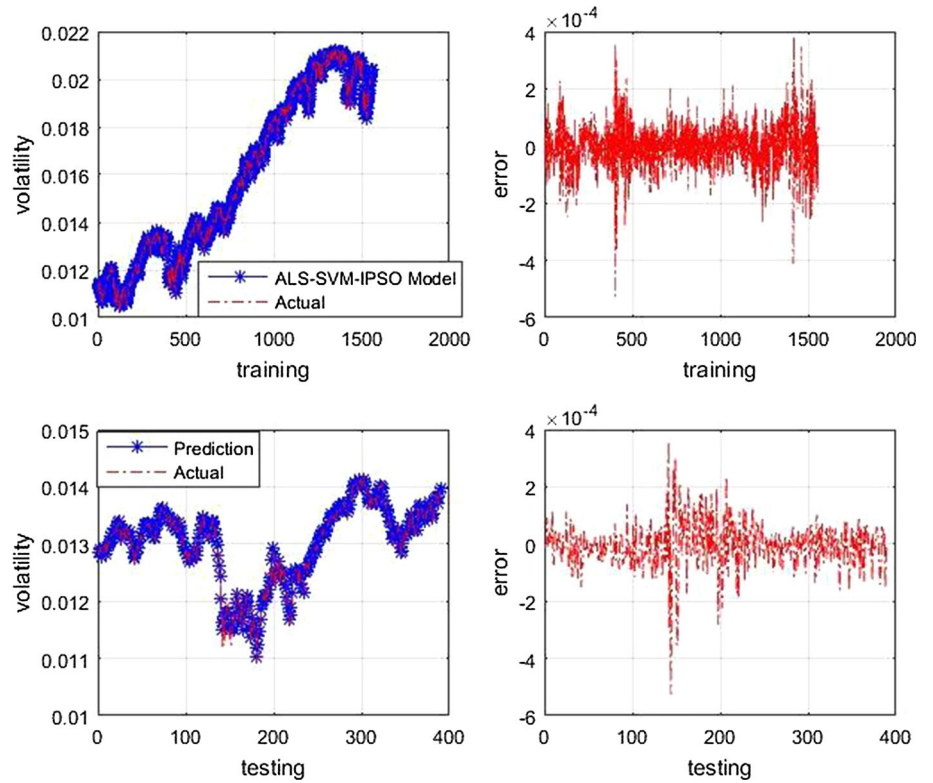
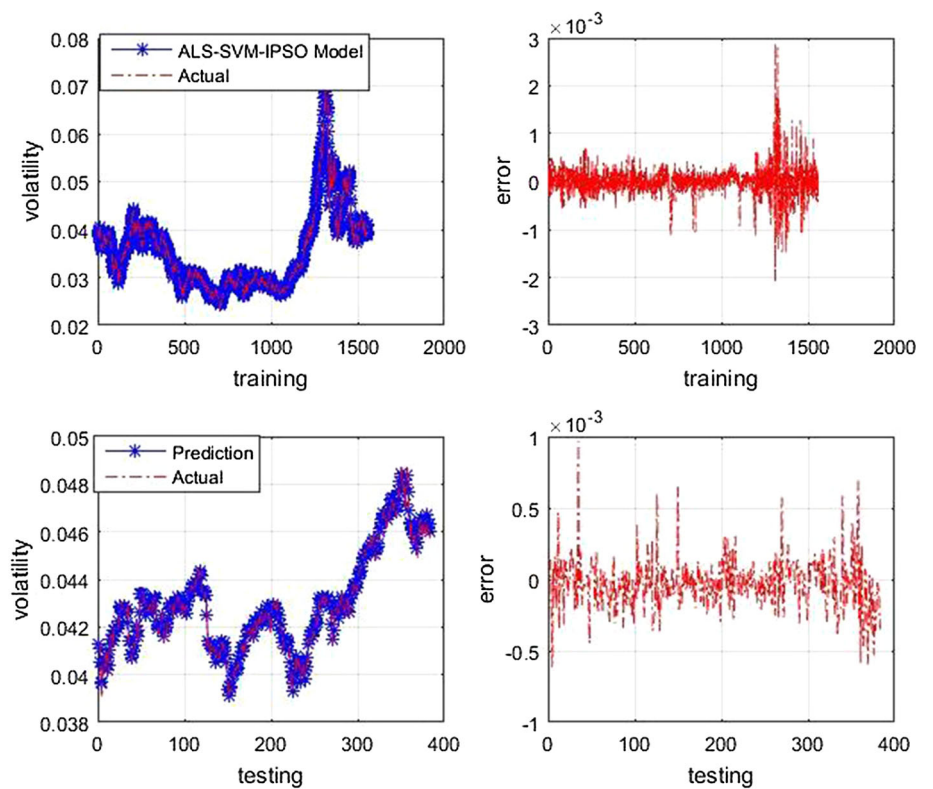


Fig. 6 The volatility forecasting and estimation errors of HS300 index



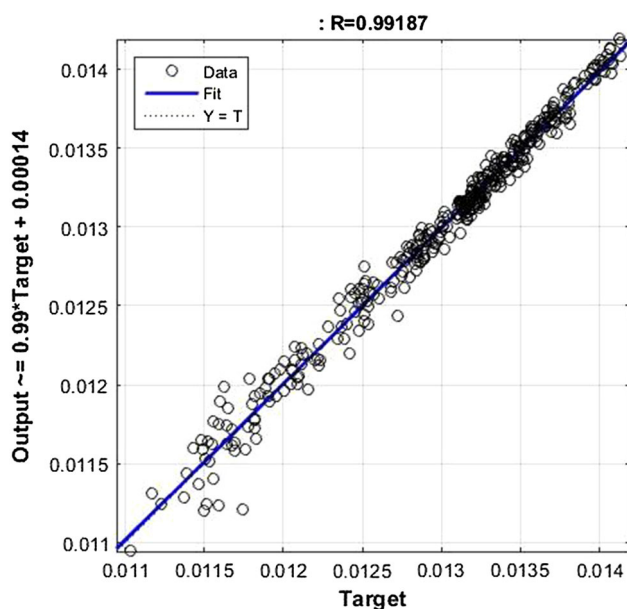


Fig. 7 The scatter plot of goodness of fitting for SP500

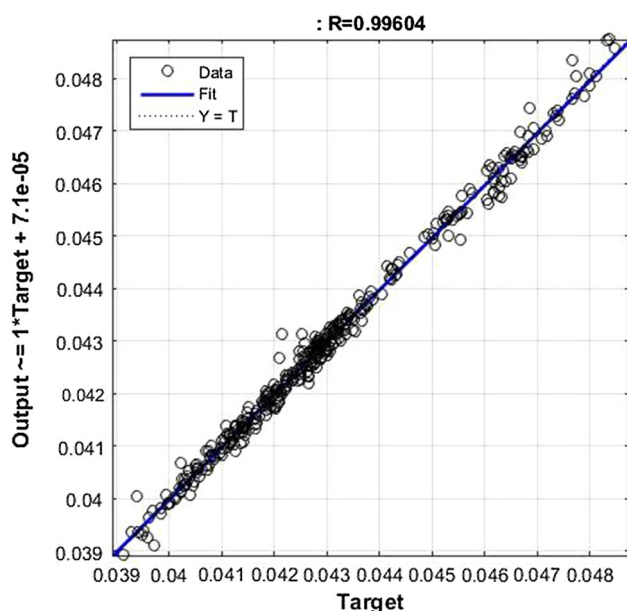


Fig. 8 The scatter plot of goodness of fitting for HS300

both training samples and testing samples, meaning that the proposed model equips with fine approximation capability and generalization ability.

The similar conclusion can also be inferred from the scatter graphs of goodness of fitting for volatility forecasting of SP500 and HS300 stock indices in Figs. 7, 8. Obviously seen from the scatter plot, it indicates that the proposed LS-SVM-IPSO approach can effectively forecast the volatility tendency of stock indexes.

8 Conclusion

Accurate forecast of stock returns volatility is challenging since the highly complex nonlinear nature of returns. In this study, we apply the LS-SVM technique optimized by the IPSO algorithm that can map nonlinear functions without prior hypothesis to forecast stock returns volatility. Furthermore, the individual GARCH family models with non-Gaussian distributions and those hybridized with the ANN method are also compared in the empirical experiments.

In terms of the loss functions, the resulting forecast performances obtained by the three different methods have been compared. It is observed that the asymmetric GARCH volatility model with generalized asymmetric Student-*t* distribution combining with ANN technique exhibits higher accuracy than individual parametric methods. The empirical studies find that the SVM technique without PSO optimization has similar modeling accuracy and generalization ability with hybrid models based on ANN. However, the LS-SVM-IPSO technique exhibits the most promising prediction performances, showing the lowest forecasting errors. Comparing with the neural network methods, the optimized least square support vector machine displays obvious higher modeling accuracy, closer approximation degree and better generalization abilities, which demonstrates to be an efficient and superior forecasting approach on the whole.

Acknowledgements We would like to acknowledge the financial support from the National Social Science Foundation of China (No. 18BGL200), the National Natural Science Foundation of China (No. 71532009), Research funding of Qingdao University (No. 41118010080).

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

Human and animals rights This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Abounoori E, Elmi Z, Nademi Y (2016) Forecasting Tehran stock exchange volatility; Markov switching GARCH approach. *Phys A* 445:264–282
- Abualigah LM, Khader AT (2017) Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. *J Supercomput* 73(11):4773–4795
- Abualigah LM, Khader AT, Hanandeh ES (2018) A new feature selection method to improve the document clustering using particle swarm optimization algorithm. *J Comput Sci* 25:456–466

- Alberg D, Shalit H, Yosef R (2008) Estimating stock market volatility using asymmetric GARCH models. *Appl Financ Econ* 18:1201–1208
- Bentes SR (2015a) A comparative analysis of the predictive power of implied volatility indices and GARCH forecasted volatility. *Phys A* 424:105–112
- Bentes SR (2015b) Forecasting volatility in gold returns under the GARCH, IGARCH and FIGARCH frameworks: new evidence. *Phys A* 424:355–364
- Bildirici M, Ersin ÖÖ (2009) Improving forecasts of GARCH family models with the artificial neural networks: an application to the daily returns in Istanbul Stock Exchange. *Expert Syst Appl* 36:7355–7362
- Bollerslev T (1986) Generalized heteroskedastic time series model for speculative prices and rates of return. *Rev Econ Stat* 69:542–547
- Cheng CH, Wei LY (2009) Volatility model based on multi-stock index for TAIEX forecasting. *Expert Syst Appl* 36:6187–6191
- Choudhary MA, Haider A (2012) Neural network models for inflation forecasting: an appraisal. *Appl Econ* 44:2631–2635
- Chuang IY, Lu JR, Lee PH (2007) Forecasting volatility in the financial markets: a comparison of alternative distributional assumptions. *Appl Financ Econ* 17:1051–1060
- Dai Q, Ma Z, Xie QY (2014) A two-phased and ensemble scheme integrated backpropagation algorithm. *Appl Soft Comput* 24:1124–1135
- Dhamija AK, Bhalla VK (2010) Financial time series forecasting: comparison of neural networks and ARCH models. *Int Res J Finance Econ* 49:185–202
- Fang F, Jonsson H, Oosterlee K, Schoutens W (2010) Fast valuation and calibration of credit default swaps under Lévy dynamics. *J Comput Finance* 14:57–86
- Göçken M, Özçalıcı M, Boru A, Dosdoğru AT (2016) Integrating metaheuristics and Artificial Neural Networks for improved stock price prediction. *Expert Syst Appl* 44:320–331
- Hajizadeh E, Seifi A, Fazel MH, Zarandi Turksen IB (2012) A hybrid modeling approach for forecasting the volatility of S&P 500 index return. *Expert Syst Appl* 39:431–436
- Ince H, Cebeci AF, Imamoglu SZ (2017) An artificial neural network-based approach to the monetary model of exchange rate. *Comput Econ* 2:1–15
- Ismail S, Shabri A, Samsudin R (2011) A hybrid model of self-organizing maps (SOM) and least square support vector machine (LSSVM) for time series forecasting. *Expert Syst Appl* 38:10574–10578
- Kristjanpoller W, Minutolo MC (2016) Forecasting volatility of oil price using an artificial neural network-GARCH model. *Expert Syst Appl* 65:233–241
- Kristjanpoller W, Fadic A, Minutolo MC (2014) Volatility forecast using hybrid Neural Network models. *Expert Syst Appl* 41:2437–2442
- Lahmiri S (2016a) Interest rate next-day variation prediction based on hybrid feedforward neural network, particle swarm optimization, and multiresolution techniques. *Phys A* 444:388–396
- Lahmiri S (2016b) A variational mode decomposition approach for analysis and forecasting of economic and financial time series. *Expert Syst Appl* 55:268–273
- Lahmiri S (2017) Modeling and predicting historical volatility in exchange rate markets. *Phys A* 471:387–395
- Liu F, Zhou ZG (2015) A new data classification method based on chaotic particle swarm optimization and least square-support vector machine. *Chem Intel Lab Syst* 147:147–156
- Lv XD, Shan X (2013) Modeling natural gas market volatility using GARCH with different distributions. *Phys A* 392:5685–5699
- Monfared SA, Enke D (2015) Noise canceling in volatility forecasting using an adaptive neural network filter. *Procedia Comput Sci* 61:80–84
- Nelson DB (1991) Conditional heteroskedasticity in asset returns: a new approach. *Econometrica* 59:347–370
- Pehlivanoglu YV (2013) A new particle swarm optimization method enhanced with a periodic mutation strategy and neural networks. *IEEE Trans Evolut Comput* 17:436–452
- Qiu M, Song Y, Akagi F (2016) Application of artificial neural network for the prediction of stock market returns: the case of the Japanese stock market. *Chaos Sol Fract* 85:1–7
- Ramyar S, Kianfar F (2017) Forecasting crude oil prices: a comparison between artificial neural networks and vector autoregressive models. *Comput Econ* 1:1–19
- Rojo-Álvarez JL, Martínez-Ramón M, Muñoz-Mari J, Camps-Valls G (2014) A unified SVM framework for signal estimation. *Digital Sign Proc* 26:1–20
- Suykens JAK, Gestel TV, Brabanter JD, Moor BD, Vandewalle J (2002) Least square support vector machines. World Scientific, Singapore
- Tseng CH, Cheng ST, Wang YH (2009) New hybrid methodology for stock volatility prediction. *Expert Syst Appl* 36:1833–1839
- Yang S, Lee J (2012) Multi-basin particle swarm intelligence method for optimal calibration of parametric Lévy models. *Expert Syst Appl* 39:482–493
- Zahedi J, Rounaghi MM (2015) Application of artificial neural network models and principal component analysis method in predicting stock prices on Tehran Stock Exchange. *Phys A* 438:178–187
- Zhao J, Yang ZJ, Xu YT (2016) Nonparallel least square support vector machine for classification. *Appl Intel* 45:1119–1128
- Zhao D, Huang C, Wei Y, Yu F, Wang M, Chen H (2017) An effective computational model for bankruptcy prediction using kernel extreme learning machine approach. *Comput Econ* 49:325–341
- Zhu DM, Galbraith JW (2010) A generalized asymmetric Student-t distribution with application to financial econometrics. *J Econometrics* 157:297–305
- Zhu DM, Galbraith JW (2011) Modeling and forecasting expected shortfall with the generalized asymmetric Student-t and asymmetric exponential power distributions. *J Emp Finance* 18:765–778

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.