Improved Gravitational Search Least Squares Support Vector Machine for Traffic Flow Prediction: Postprint

Authors: Xu Qinshuai, He Qing, Wei Kangyuan

Date: 2018-10-11T00:00:00+00:00

Abstract

In the field of intelligent transportation systems, accurate traffic flow prediction plays a significant role. To improve the accuracy of traffic flow prediction models based on least squares support vector machines, a novel improved gravitational search algorithm (TCK-AGSA) is proposed for parameter optimization. First, the Kbest function is improved based on Tent mapping, enabling the algorithm to possess a mechanism for escaping local optima; then, a global optimal guidance strategy is introduced to accelerate particles toward the optimal solution; subsequently, evolutionary degree factor and aggregation degree factor are introduced into the velocity update weight coefficient, endowing the algorithm with strong adaptive capability. Simulation results on 12 benchmark functions demonstrate that TCK-AGSA outperforms GSA and its improved variants. Finally, a least squares support vector machine model optimized by TCK-AGSA is established, and real highway traffic flow data from Guizhou Province in 2016 is selected for prediction experiments. The results show that the model possesses better prediction accuracy, robustness, and generalization ability.

Full Text

Traffic Flow Forecasting Using Least Squares Support Vector Machine Optimized by Modified Gravitational Search Algorithm

Xu Qinshuai[1 2], He Qing[1 2]†, Wei Kangyuan[1 2] [1]College of Big Data & Information Engineering, Guizhou University [2]Guizhou Provincial Key Laboratory of Public Big Data, Guiyang 550025, China

Abstract

Accurate traffic flow forecasting plays a crucial role in the field of Intelligent Transportation Systems. To improve the precision of traffic flow prediction models based on Least Squares Support Vector Machine (LSSVM), this paper proposes a novel modified gravitational search algorithm (TCK-AGSA) for parameter optimization. First, the Kbest function is improved based on Tent mapping, enabling the algorithm to escape local optima. Second, a global optimal guidance strategy is introduced to accelerate particle movement toward the optimal solution. Finally, evolutionary and convergence factors are incorporated into the velocity update weight coefficient to enhance the algorithm' s adaptive capability. Simulation results on twelve benchmark functions demonstrate that TCK-AGSA outperforms GSA and its variants. Furthermore, an LSSVM model optimized by TCK-AGSA is established and tested using real highway traffic flow data from Guizhou Province in 2016. The results show that the proposed model achieves better prediction accuracy, robustness, and generalization capability.

Keywords: gravitational search algorithm; chaos optimization algorithm; adaptive weighted coefficient; least squares support vector machine; traffic flow forecasting

0 Introduction

With rapid economic development and increasing vehicle ownership, transportation networks face mounting pressure from traffic congestion, accidents, and air pollution. Intelligent Transportation Systems (ITS) offer effective solutions to these critical traffic problems through advanced traffic guidance control and monitoring technologies. As an essential component of ITS, traffic flow prediction has attracted widespread attention in recent decades.

Accurate traffic flow forecasting is vital for analyzing road conditions, improving network planning, and optimizing traffic control strategies. To enhance prediction accuracy, numerous scholars have developed various forecasting models that generally fall into two categories:

a) **Classical statistical models**: These models fit and predict traffic flow data through regression and parameter optimization. For instance, Kumar et al. proposed a seasonal SARIMA model based on ARIMA to account for seasonal traffic variations, addressing the applicability issues of basic ARIMA models. Subsequently, Kumar introduced a Kalman filtering-based prediction model in 2017, overcoming SARIMA' s dependence on large datasets.

b) **Data-driven models**: These models leverage intelligent computational methods to mine evolutionary trends from historical data. In deep learning, Lv et al. proposed a stacked autoencoder hybrid model that significantly improved prediction accuracy. Polson et al. developed a linear deep learning model combining L1 regularization and tanh layers to handle sharp nonlinear transitions in traffic states. In support vector machine applications, Hu et al. introduced a PSO-SVR model optimized by particle swarm optimization, incorporating historical traffic flow momentum to reduce noise impact. Kang et al. developed an online LSSVM-based method that shortened update times while maintaining accuracy. Shang et al. proposed a phase-space reconstruction and combined kernel LSSVM model optimized by PSO, demonstrating strong prediction capability and robustness. Cong et al. introduced an FOA-LSSVM model using fruit fly optimization, which outperformed PSO-LSSVM and RBFNN models.

Among data-driven approaches, LSSVM shows superior performance in traffic flow prediction due to short training times and strong generalization. However, the selection of regularization parameter C and kernel width σ critically affects model complexity, accuracy, and generalization capability. While swarm intelligence algorithms like PSO, cuckoo search, and ant colony optimization have been applied to LSSVM parameter optimization, achieving ideal accuracy remains challenging.

The Gravitational Search Algorithm (GSA), proposed by Rashedi et al. in 2009 based on Newton' s law of universal gravitation, has demonstrated better convergence than PSO and genetic algorithms. However, basic GSA suffers from premature convergence, local optima entrapment, and lack of memory functionality. Consequently, researchers have proposed various improvements: Darzi et al. introduced individual local best concepts from PSO to address memory deficiency; Zhang et al. enhanced GSA with adaptive gravitational coefficient decay, elite memory guidance, and mutation operations; Guo et al. developed an information entropy-based approach with dynamic weight selection. Despite these advances, balancing exploration and exploitation while improving escape from local optima remains challenging.

Therefore, this paper proposes a Tent map-based chaotic Kbest adaptive GSA (TCK-AGSA). The chaotic Kbest strategy enables escape from local optima, while global optimal guidance accelerates convergence. Adaptive weight coefficients based on evolutionary and convergence factors enhance adaptability. Experimental validation on twelve benchmark functions and real traffic data from Guizhou highways demonstrates the algorithm' s superiority.

1 Related Work

1.1 Least Squares Support Vector Machine

LSSVM optimizes and extends the SVM mathematical model by replacing inequality constraints with equality constraints and using an L2-norm error function as the loss function, transforming the QP problem into a linear system solution.

Given a training set with N samples $\{(x_i, y_i)\}_{i=1}^N$, where $x_i \in \mathbb{R}^d$ is the input vector and $y_i \in \mathbb{R}$ is the output vector, LSSVM maps samples to a high-dimensional space using nonlinear mapping function $\varphi(x)$ to construct the linear regression function:

$$y(x) = w^T \varphi(x) + b$$

where w is the weight vector and b is the bias.

Based on structural risk minimization, the LSSVM regression optimization problem is:

$$\min_{w,b,\xi} J(w,\xi) = \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^{N} \xi_i^2$$

subject to:

$$y_i = w^T \varphi(x_i) + b + \xi_i, \quad i = 1, \dots, N$$

where C is the regularization parameter and ξ_i are error variables.

The Lagrangian function is constructed as:

$$L(w,b,\xi,\alpha) = J(w,\xi) - \sum_{i=1}^{N} \alpha_i \left(w^T \varphi(x_i) + b + \xi_i - y_i \right)$$

where α_i are Lagrange multipliers.

From the KKT conditions, we obtain:

$$\begin{cases} \frac{\partial L}{\partial w} = 0 \to w = \sum_{i=1}^{N} \alpha_i \varphi(x_i) \\ \frac{\partial L}{\partial b} = 0 \to \sum_{i=1}^{N} \alpha_i = 0 \\ \frac{\partial L}{\partial \xi_i} = 0 \to \alpha_i = C\xi_i \\ \frac{\partial L}{\partial \alpha_i} = 0 \to w^T \varphi(x_i) + b + \xi_i - y_i = 0 \end{cases}$$

Eliminating w and ξ yields:

$$\begin{bmatrix} 0 & 1^T \\ 1 & K + C^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}$$

where $y = [y_1, \dots, y_N]^T$, 1 is an all-ones vector, I is the identity matrix, and K is the kernel function matrix with $K_{ij} = \varphi(x_i)^T \varphi(x_j) = k(x_i, x_j)$.

The final LSSVM regression function is:

$$y(x) = \sum_{i=1}^{N} \alpha_i k(x, x_i) + b$$

The Gaussian Radial Basis Function (RBF) kernel is adopted:

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

where σ is the kernel bandwidth.

The regularization parameter C and kernel bandwidth σ directly affect learning and generalization capabilities. C balances structural and empirical risk—large values reduce empirical risk but may cause overfitting, while small values reduce complexity but may cause underfitting. Similarly, large σ values produce smoother kernels (potential underfitting), while small values produce steeper kernels (potential overfitting). Therefore, selecting appropriate parameters is crucial, and this paper employs the improved GSA for LSSVM parameter optimization.

1.2 Gravitational Search Algorithm

In GSA, each particle is treated as a mass object moving through the search space according to Newton' s second law, influenced by gravitational forces from other particles.

A population of size N in a D-dimensional search space has particles with positions $X_i = (x_i^1, \ldots, x_i^d, \ldots, x_i^D)$. At iteration t, particle i' s inertial mass $M_i(t)$ is calculated based on its fitness:

$$M_i(t) = \frac{f_i(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)}$$

where $f_i(t)$ is particle i' s fitness at iteration t, $\text{best}(t) = \min_{j\in\{1,\ldots,N\}} f_j(t)$, and $\text{worst}(t) = \max_{j\in\{1,\ldots,N\}} f_j(t)$ for minimization problems (reversed for maximization).

The gravitational force on particle i from particle j in dimension d is:

$$F_{ij}^d(t) = G(t)\frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon}\left(x_j^d(t) - x_i^d(t)\right)$$

where $G(t)$ is the gravitational coefficient, ε is a small constant, and $R_{ij}(t)$ is the Euclidean distance between particles.

The gravitational coefficient decays over time:

$$G(t) = G_0 \times e^{-\alpha t/T}$$

where G_0 is the initial constant, α is the decay coefficient, and T is the maximum iteration number.

The total force on particle i in dimension d is:

$$F_i^d(t) = \sum_{j \in \text{Kbest}, j \neq i} \text{rand}_j \times F_{ij}^d(t)$$

where Kbest is the set of elite particles with best fitness and largest masses, initially N and decreasing over time, and rand_j is a random number in $[0, 1]$.

Particle acceleration is:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)}$$

Velocity and position are updated as:

$$v_i^d(t+1) = \text{rand}_i \times v_i^d(t) + a_i^d(t)$$
$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1)$$

where rand_i is a random number in $[0, 1]$.

Basic GSA has three main limitations: (a) decreasing population diversity may cause premature convergence by losing potentially good particles with small masses; (b) the algorithm lacks memory and information sharing mechanisms, potentially losing previous best results and accelerating movement toward inferior positions; (c) there is no mechanism to escape local optima. These deficiencies motivate the proposed improvements.

2 Improved Gravitational Search Algorithm Design

2.1 Chaotic Kbest Based on Tent Mapping

In GSA, Kbest represents the percentage of elite particles exerting forces on others. The linear Kbest function is:

$$\text{Kbest}(t) = \text{final\_per} + \left(1 - \frac{t}{T}\right) \times (1 - \text{final\_per})$$

where $\text{final\_per} \in [0.9, 1]$.

To address local optima entrapment, chaotic mapping' s randomness and ergodicity are employed to make Kbest decrease nonlinearly. While CKGSA used

Logistic mapping, its non-uniform distribution in $[0,1]$ (higher probability near boundaries) affects efficiency. Tent mapping, a one-dimensional piecewise linear map, provides more uniform distribution, faster iteration, and better optimization efficiency:

$$x_{t+1} = \begin{cases} 2x_t, & 0 \le x_t \le 0.5 \\ 2(1-x_t), & 0.5 < x_t \le 1 \end{cases}$$

where $x_t \in [0,1]$.

The chaotic Kbest function based on Tent mapping is:

$$\text{Kbest}(t) = \text{final\_per} + x_t \times (1 - \text{final\_per}) \times \left(1 - \frac{t}{T}\right) \times N$$

where N is the population size, T is the maximum iteration number, and x_t is the chaotic number at iteration t.

2.2 Adaptive Weight Global Optimal Guided Velocity Update

To address GSA’ s lack of memory and optimal value oscillation, the velocity update is modified as:

$$v_i^d(t+1) = \omega \cdot v_i^d(t) + c_1 \cdot \text{rand}_i \cdot a_i^d(t) + c_2 \cdot \text{rand}_j \cdot (\text{gbest}^d(t) - x_i^d(t))$$

where ω is the velocity weight coefficient, c_1 and c_2 are individual and social learning factors, rand_i and rand_j are random numbers in $[0,1]$, and $\text{gbest}^d(t)$ is the global best position.

The weight coefficient ω significantly affects optimization performance. Large ω enhances global exploration, while small ω facilitates local exploitation. An adaptive weight based on evolutionary and convergence factors is proposed.

At iteration t, define particle i’ s fitness as $f(x_i(t))$, its personal best as $f(\text{pbest}_i(t))$, and the global best as $f(\text{gbest}(t))$.

The evolutionary factor is:

$$\text{evot}_i(t) = a_1 \cdot \frac{f(\text{pbest}_i(t-1)) - f(\text{pbest}_i(t))}{|f(\text{pbest}_i(t-1)) - f(\text{gbest}(t))|} + a_2 \cdot \frac{f(\text{pbest}_i(t)) - f(\text{gbest}(t))}{|f(\text{pbest}_i(t)) - f(\text{gbest}(t))|} + a_3$$

where $a_1, a_2, a_3 \in [0,1]$ with $a_1 + a_2 + a_3 = 1$ and $\max(a_1, a_2, a_3) > 0.5$.

The average personal best fitness is:

$$\text{ave\_fpbest}(t) = \frac{1}{N}\sum_{i=1}^{N} f(\text{pbest}_i(t))$$

The convergence factor is:

$$\text{cont}(t) = \frac{\text{ave\_}f(x(t))}{\text{ave\_fpbest}(t)}$$

where $\text{ave\_}f(x(t))$ is the average fitness of all particles.

When the evolutionary factor is large (poor evolution), ω should decrease to enhance local exploitation. When the convergence factor is large (high aggregation), ω should increase to enhance global exploration. The adaptive weight coefficient is:

$$\omega_i(t) = \omega_0 \cdot (b_1 \cdot \text{evot}_i(t) + b_2 \cdot \text{cont}(t))$$

where ω_0 is the initial weight, and b_1, b_2 are balance coefficients with $b_1, b_2 \in (0, 1)$ and $b_1 + b_2 = 1$.

The complete velocity update equation becomes:

$$v_i^d(t+1) = \omega_i(t) \cdot v_i^d(t) + c_1 \cdot \text{rand}_i \cdot a_i^d(t) + c_2 \cdot \text{rand}_j \cdot (\text{gbest}^d(t) - x_i^d(t))$$

3 Simulation Experiments and Results Analysis

Experiments were conducted in MATLAB R2013a on a 64-bit Windows 7 system. Twelve benchmark functions were selected for numerical experiments, comparing TCK-AGSA with basic GSA, CKGSA [24], and GG-GSA [25].

Table 1 shows parameter settings. For fairness, population size $N = 50$, maximum iterations $T = 1000$, and initial gravitational constant $G_0 = 100$ were used for GSA and TCK-AGSA. Other parameters follow the literature [24,25].

Table 2 lists the twelve benchmark functions: F1-F4 are unimodal (F4 is a discontinuous step function), F5-F8 are high-dimensional multimodal, and F9-F12 are low-dimensional multimodal. Each function ran independently 30 times to reduce random interference.

Table 3 compares optimization results. Mean, Best, and Std.Dev represent the average, best, and standard deviation over 30 runs. Results for CKGSA and GG-GSA are from their respective papers; "-" indicates unavailable data. Bold values indicate the best results.

For unimodal functions F1-F3, TCK-AGSA significantly outperforms others in convergence precision and stability. For F4, all algorithms achieve the theoretical optimum, showing GSA’ s effectiveness for discontinuous functions.

For high-dimensional multimodal functions F5-F8, TCK-AGSA shows superior performance on F6, F7, and F8, though slightly less precise than GG-GSA on F5. For F8 (Griewank function), which has numerous local optima, TCK-AGSA achieves the theoretical optimum, demonstrating the chaotic Kbest strategy’ s effectiveness in escaping local optima.

For low-dimensional multimodal functions F9-F12, TCK-AGSA exhibits better convergence precision and speed. All three improved algorithms achieve theoretical optima for F11 and F12, but TCK-AGSA’ s mean and standard deviation are superior, indicating better robustness.

Figure 1 [Figure 1: see original paper] shows convergence curves for six selected functions, illustrating TCK-AGSA’ s improved convergence speed and precision over basic GSA, particularly for F3 where improvement reaches seven orders of magnitude.

Overall, TCK-AGSA effectively handles complex optimization problems, demonstrating superior convergence precision, speed, and robustness compared to GSA, CKGSA, and GG-GSA.

4 Traffic Flow Prediction Experiment

4.1 Data Foundation and Preprocessing

The dataset comprises highway traffic flow data from Guizhou Province, collected at observation station D534 in November 2016. It includes 30 dimensions such as vehicle counts, speed, lane numbers, and average headway. After cleaning and dimensionality reduction, data from November 7-11 (5 days) for lanes 11, 12, 31, and 32 were selected, with 15-minute intervals yielding 96 daily samples. The weekday data exhibits clear periodicity. Data from November 7-10 (Monday-Thursday) served as the training set, while November 11 (Friday) was used for testing.

To address instability and large variations in real traffic data, normalization was applied:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

where x is the original value, x' is the normalized value, and $x_{\max}$, $x_{\min}$ are the maximum and minimum flow values.

4.2 Traffic Flow Prediction Using TCK-AGSA Optimized LSSVM Model

The TCK-AGSA-LSSVM prediction model uses real traffic data as input and outputs optimized parameters C and σ along with predictions. The implementation flowchart is shown in Figure 2 [Figure 2: see original paper].

The algorithm proceeds as follows: a) Initialize TCK-AGSA parameters (particle number N, maximum iterations T, etc.) and randomly deploy particles. Initialize LSSVM parameter ranges and map them to particle positions. b) Use Mean Squared Error (MSE) between predictions and actual values as the fitness function. c) Update inertial masses using equations (12) and (13), and gravitational coefficient using equation (17). d) Calculate inter-particle attraction forces using equation (16). e) Compute chaotic Kbest using equation (25) and resultant forces using equation (19). f) Calculate acceleration using equation (20) and adaptive weight coefficient using equation (31). g) Update velocity using equation (32) and position using equation (22). h) Check termination criteria; if not met, return to step b. i) Map the global best particle position to optimal LSSVM parameters C and σ. j) Train the optimized model and output predictions, evaluating performance through prediction errors.

4.3 Reference Models and Evaluation Metrics

To validate effectiveness, TCK-AGSA-LSSVM is compared with basic LSSVM, FOA-LSSVM [10], and GSA-LSSVM [14]. For fair comparison, population size $N = 50$, maximum iterations $T = 100$, and dimension $D = 2$ were used for all optimization algorithms. FOA step size was set to 10; GSA and TCK-AGSA used parameters from Table 1. Basic LSSVM used $C = 500$ and $\sigma = 0.1$.

Evaluation metrics include: - Mean Squared Error: $\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$ - Mean Absolute Percentage Error: $\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i-\hat{y}_i}{y_i}\right| \times 100\%$ - Mean Absolute Error: $\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$ - Fitting Coefficient: $\mathrm{EC} = 1 - \frac{\sum_{i=1}^{N}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{N}(y_i-\bar{y})^2}$

Lower MSE, MAPE, and MAE indicate better performance; EC closer to 1 indicates better fit.

4.4 Comparative Analysis of Prediction Results

Experiments used MATLAB R2013a with the LSSVMlab1.8 toolbox. Table 4 shows parameter optimization results for different models. Figures 3 [Figure 3: see original paper] through 6 [Figure 6: see original paper] compare prediction effects, where i represents time index and q represents traffic flow.

The TCK-AGSA-LSSVM prediction curve best matches the actual data trend (Figures 3a-6a). Scatter plots (Figures 3b-6b) show TCK-AGSA-LSSVM has the smallest deviation from actual values. Relative error distribution (Figure

7 [Figure 7: see original paper]) shows TCK-AGSA-LSSVM maintains most errors below 5% with the smallest fluctuation range, demonstrating improved accuracy even at difficult prediction points.

Table 5 compares evaluation metrics. TCK-AGSA-LSSVM achieves the lowest MSE, MAPE, and MAE values while attaining the highest EC values, confirming its superior prediction accuracy and fit.

5 Conclusion

This paper proposes TCK-AGSA, an improved gravitational search algorithm featuring: (1) Tent map-based chaotic Kbest for escaping local optima, (2) global optimal guidance for accelerated convergence, and (3) adaptive weight coefficients based on evolutionary and convergence factors for better exploration-exploitation balance. Benchmark tests on twelve functions demonstrate superior performance over GSA, CKGSA, and GG-GSA in convergence precision, speed, and robustness.

The TCK-AGSA-LSSVM traffic flow prediction model, tested on 2016 Guizhou highway data, achieves lower MSE, MAPE, and MAE while obtaining higher EC values compared to LSSVM, FOA-LSSVM, and GSA-LSSVM models. This validates the proposed method' s superior accuracy, robustness, and generalization capability, providing valuable support for traffic guidance and control applications.

References

[1] Kalamaras I, Zamichos A, Salamanis A, et al. An interactive visual analytics platform for smart intelligent transportation systems management [J]. IEEE Trans on Intelligent Transportation Systems, 2018, 19 (2): 1-10.

[2] Rajabzadeh Y, Rezaie A H, Amindavar H. Short-term traffic flow prediction using time-varying Vasicek model [J]. Transportation Research Part C, 2017, 74: 168-181.

[3] Kumar S V, Vanajakshi L. Short-term traffic flow prediction using seasonal ARIMA model with limited input data [J]. European Transport Research Review, 2015, 7 (3): 21.

[4] Kumar S V. Traffic flow prediction using Kalman filtering technique [J]. Procedia Engineering, 2017, 187: 582-587.

[5] Lv Yisheng, Duan Yanjie, Kang Wenwen, et al. Traffic flow prediction with big data: a deep learning approach [J]. IEEE Trans on Intelligent Transportation Systems, 2015, 16 (2): 865-873.

[6] Polson N G, Sokolov V O. Deep learning for short-term traffic flow prediction [J]. Transportation Research Part C: Emerging Technologies, 2017, 79: 1-17.

[7] Hu Wenbin, Yan Liping, Liu Kaizeng, et al. A short-term traffic flow forecasting method based on the hybrid PSO-SVR [J]. Neural Processing Letters, 2016, 43 (1): 155-172.

[8] Kang Jun, Duan Zongtao, Tang Lei, et al. LS-SVM online short term traffic flow prediction method [J/OL]. Application Research of Computers, 2018, 35 (10). http://www.arocmag.com/article/02-2018-10-022.html.

[9] Shang Qiang, Lin Ciyun, Yang Zhaosheng, et al. Short-term traffic flow prediction model using particle swarm optimization-based combined kernel function-least squares support vector machine combined with chaos theory and its application [J]. Advances in Mechanical Engineering, 2016, 8 (8): 1-12.

[10] Cong Yuliang, Wang Jianwei, Li Xiaolei. Traffic flow forecasting by a least squares support vector machine with a fruit fly optimization algorithm [J]. Procedia Engineering, 2016, 137: 59-68.

[11] Suykens J A K, Vandewalle J. Least squares support vector machine classifiers [J]. Neural Processing Letters, 1999, 9 (3): 293-300.

[12] Wang Xianfang, Zhang Yue, Wang Junmei. Predicting protein model quality assessment by SAPSO-LSSVM [J]. Application Research of Computers, 2017, 34 (5): 1346-1348+1378.

[13] Geng Liyan, Chen Lihua. Forecast on railway traffic volume using mixed-kernel LSSVM optimized by FOA [J]. Application Research of Computers, 2017, 34 (2): 409-412.

[14] Yuan Xiaohui, Chen Chen, Yuan Yanbin, et al. Short-term wind power prediction based on LSSVM–GSA model [J]. Energy Conversion & Management, 2015, 101: 393-401.

[15] Sun Wei, Sun Jingyi. Daily PM2.5 concentration prediction based on principal component analysis and LSSVM optimized by cuckoo search algorithm [J]. Journal of Environmental Management, 2016, 188: 144-152.

[16] Liu Chao, Niu Peifeng, Li Guoqiang, et al. A hybrid heat rate forecasting model using optimized LSSVM based on improved GSA [J]. Neural Processing Letters, 2017, 45 (1): 299-318.

[17] Li Yanan, Yang Peng, Wang Huajun. Short-term wind speed forecasting based on improved ant colony algorithm for LSSVM [J]. Cluster Computing, 2018, 9: 1-7.

[18] Kumar L, Sripada S K, Sureka A, et al. Effective fault prediction model developed using least square support vector machine (LSSVM) [J]. Journal of Systems & Software, 2018, 137: 686-712.

[19] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm [J]. Information Sciences, 2009, 179 (13): 2232-2248.

[20] Li Chaoshun, Zhou Jianzhong. Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm [J]. Energy Conversion and Management, 2011, 52 (1): 374-381.

[21] Darzi S, Kiong T S, Islam M T, et al. A memory-based gravitational search algorithm for enhancing minimum variance distortionless response beamforming [J]. Applied Soft Computing, 2016, 47 (C): 103-118.

[22] Zhang Nan, Li Chaoshun, Li Ruhai, et al. A mixed-strategy based gravitational search algorithm for parameter identification of hydraulic turbine governing system [J]. Knowledge-Based Systems, 2016, 109: 218-234.

[23] Guo Jiehao, Gao Xingbao. Hybrid gravitational search algorithm with information entropy [J]. Application Research of Computers, 2016, 33 (05): 1319-1321, 134.

[24] Mittal H, Pal R, Kulhari A, et al. Chaotic Kbest gravitational search algorithm (CKGSA) [C]// Proc of the 9th International Conference on Contemporary Computing. Piscataway, NJ: IEEE Press, 2017: 1-6.

[25] Bohat V K, Arya K V. An effective gbest-guided gravitational search algorithm for real-parameter optimization and its application in training of feedforward neural networks [J]. Knowledge-Based Systems, 2018, 143: 236-257.

Note: Figure translations are in progress. See original paper for figures.