Efficient Network Traffic Prediction Method Based on PF-LSTM Network Postprint

Authors: Li Xiaolin, Wu Teng

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

Abstract

Network traffic exhibits real-time burstiness and dependency, and traditional network traffic prediction models suffer from weaknesses such as poor generalization capability and low prediction accuracy. To address this problem, a network traffic prediction model based on Long Short-Term Memory (LSTM) recurrent neural network is proposed. First, the resampling process of the Particle Filter (PF) algorithm is improved using distance comparison and optimization combination strategies; then a PF-LSTM network-based network traffic prediction model is constructed, where the improved PF algorithm is applied to model training to enhance its training speed and overcome the drawback of traditional LSTM networks converging to local optima; finally, the proposed model is applied to network traffic prediction. Experimental results demonstrate that, compared with the traditional LSTM model, the proposed PF-LSTM model exhibits superior prediction accuracy and convergence efficiency, and can better characterize the variation trends of network traffic.

Full Text

Preamble

Title: Efficient Network Traffic Prediction Method Based on PF-LSTM Network

Authors: Li Xiaolin[1],[2], Wu Teng[1]
[1]Chongqing University of Posts and Telecommunications, Research Center for Communication New Technology Application, Chongqing 400065, China
[2]Chongqing Information Technology Designing Co., Ltd, Chongqing 401121, China

Abstract: Network traffic exhibits real-time sudden changes and dependencies, and traditional network traffic prediction models suffer from weak generalization ability and low prediction accuracy. To address these issues, this paper proposes a network traffic prediction model based on Long Short-Term Memory (LSTM) recurrent neural networks. First, we improve the resampling process of the Particle Filter (PF) algorithm using distance comparison and optimized combination strategies. Then, we construct a PF-LSTM network model for network traffic prediction, employing the improved PF algorithm for model training to enhance training speed and overcome the local optimum convergence drawback of traditional LSTM networks. Finally, we apply the proposed model to network traffic prediction. Experimental results demonstrate that compared with the traditional LSTM model, the proposed PF-LSTM model achieves better prediction accuracy and convergence efficiency, and can more effectively describe network traffic variation trends.

Keywords: network traffic prediction; long short-term memory neural network; particle filter algorithm; prediction model

0 Introduction

With the rapid development of communication technologies, wireless network users have increased dramatically, and network traffic has grown explosively. Effective management of wireless networks has become crucial for preventing network congestion and improving resource utilization. Modeling and prediction of network traffic can help us understand traffic variation characteristics and trends in advance, enabling the formulation of reasonable and effective traffic management strategies to meet users' Quality of Service (QoS) requirements. Therefore, establishing high-precision prediction models holds significant practical importance.

Traditional linear models for short-term network traffic prediction include Autoregressive (AR), Moving Average (MA), Autoregressive Moving Average (ARMA), and Autoregressive Integrated Moving Average (ARIMA) models [1-2]. While these models have achieved good results for small-scale sparse network traffic prediction, they cannot satisfy the nonlinear requirements of modern network traffic, such as sudden changes and multiple dependencies [3]. With the continuous emergence of neural networks and support vector machines, machine learning-based prediction models have been developed, including Artificial Neural Networks, Least Squares Support Vector Machine (LSSVM), and Extreme Learning Machine (ELM) [4-6]. By fitting and learning regular patterns from historical data, these models better explain the randomness and periodicity of network traffic and have been widely applied. However, these algorithms lack consideration of temporal correlations in time-series data, resulting in limited prediction accuracy [7].

Recurrent Neural Network (RNN) is a deep neural network that introduces recurrent feedback [8], taking into account the temporal correlation of time series and demonstrating stronger practicality in learning long-term dependent temporal data [9]. Long Short-Term Memory (LSTM) recurrent neural network is a special RNN model that can learn long-term dependencies in sequential data and effectively solve the gradient vanishing and gradient explosion problems in conventional RNN training, making it widely used in time series prediction [10].

Traditional LSTM determines network parameters using the Back Propagation Through Time (BPTT) algorithm, which has extremely high computational complexity and tends to converge to local optimal solutions [11]. Particle Filter (PF) is a Bayesian estimation algorithm based on Monte Carlo methods, exhibiting stronger modeling capabilities in nonlinear, non-Gaussian systems. The algorithm features fast convergence to global optimal solutions and has been successfully applied in multi-target tracking, fault diagnosis, signal processing, and other fields [12-14]. However, the basic PF algorithm suffers from particle degeneracy and loss of particle diversity [15].

To address the LSTM parameter optimization problem in network traffic prediction, this paper proposes an improved PF-LSTM network traffic prediction model that uses an improved PF algorithm to determine optimal model parameters, thereby improving prediction accuracy and efficiency.

1.2 LSTM Neural Network Principle

LSTM, as an improved RNN model, is essentially similar in principle but differs by introducing a "memory cell" structure in the hidden layer and using different functions to calculate hidden layer states. In the "memory cell," three gating layers control the amount of information that can pass through, making it highly effective for sequential data with long-term dependencies. The structure of the LSTM "memory cell" is shown in Figure 2 [Figure 2: see original paper].

In Figure 2, i_t represents the input gate that controls information input; f_t represents the forget gate that controls retention of the cell' s historical state; o_t represents the output gate that controls information output. The activation function σ is the sigmoid function applied in the three gates, producing values between [0,1]. When the forget gate outputs 0, it means all information from the previous state is discarded; when the forget gate outputs 1, it means all information from the previous state is retained. The specific calculation formulas for the LSTM structure are:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{3}$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{4}$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{5}$$
$$\tilde{c}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{6}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{7}$$
$$h_t = o_t \odot \tanh(c_t) \tag{8}$$
$$y_t = \sigma(W_{hy}h_t) \tag{9}$$

where x_t is the input layer input, i_t is the input gate output, f_t is the forget gate output, o_t is the output gate output, $\tilde{c}_t$ is used to update the cell state, h_t is the hidden layer output, and y_t is the output layer output. W represents the weight matrix for corresponding layers, b represents the bias vector for each layer output, and σ and tanh are activation functions, namely the sigmoid function and hyperbolic tangent function, respectively.

The BPTT algorithm, similar in principle to Back Propagation (BP), can be roughly divided into four steps: (a) calculate LSTM cell output values according to equations (3)-(8); (b) compute error terms for each LSTM cell in two backpropagation directions: through time and network hierarchy; (c) calculate gradients for each weight based on corresponding error terms; (d) update weights using gradient descent optimization algorithms. During training, error terms are passed upward to previous states with almost no attenuation. States from long before and the final state both affect weight adjustments across layers, resulting in a trained model with long-term memory capabilities. However, the BPTT algorithm requires derivative operations, imposing strict continuity and differentiability requirements on the optimization function and constraints. In complex optimization environments, it has high computational complexity and may converge to local optimal values.

Due to improved modern computer computational efficiency in recent years, particle filters have demonstrated excellent superiority in nonlinear, non-Gaussian systems and have been successfully applied in intelligent navigation and tracking, data analysis, fault detection, and other fields. To improve LSTM model prediction accuracy, reduce computational complexity, and enhance efficiency, this paper introduces an improved particle filter nonlinear constrained optimization algorithm to achieve LSTM prediction model parameter optimization through particle filtering.

1.1 Recurrent Neural Network Principle

Deep learning, as a deep-level neural network, is a new research field in machine learning. Traditional artificial neural networks consider data to be independent, with the only input being current moment input data, without considering temporal relationships between data. RNN is a feedforward neural network that adds memory units, where hidden layer nodes are no longer unconnected but connected.

Figure 1 [Figure 1: see original paper] shows a typical RNN unit, including input layer x, hidden layer h, and output layer o. One unidirectional information flow goes from input layer to hidden layer, then from hidden layer to output layer, forming the basic neural network structure. Another information flow goes from hidden layer to hidden layer, forming a closed loop and creating self-connected hidden layers. The RNN principle is as follows: x_t represents input at time t, h_t represents hidden layer state at time t, and o_t represents output. The current hidden layer state contains not only input layer information but also previous moment information h_{t-1}, while h_t also affects the hidden layer at next moment h_{t+1}. The RNN unit structure can be expressed using formulas:

$$h_t = \tanh(Ux_t + Wh_{t-1}) \tag{1}$$

$$o_t = \text{softmax}(Vh_t) \tag{2}$$

where U, W, and V are weight matrices for corresponding layers.

The structural characteristics of RNN hidden layers enable recurrent neural networks to "remember" previous information and apply it to current outputs, thus effectively solving long-term dependency problems. However, as the time interval increases continuously, RNN loses its ability to learn information from the distant past, namely the gradient vanishing problem.

2.1 Standard Particle Filter Algorithm

The particle filter algorithm is a Bayesian estimation method based on Monte Carlo ideas. Its fundamental concept is to replace integration operations with weighted summation in state space, approximating the posterior probability density $p(x_k|z_{1:k})$ of random signal x_k (where $z_{1:k}$ represents observations), and then continuously updating through system observation function $h(\cdot)$ to obtain the minimum variance estimate of the signal. That is:

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_k - x_k^i) \tag{10}$$

where w_k^i represents particle weights and N represents the number of sampling particles.

The standard particle filter algorithm includes key steps such as importance sampling, resampling, and state estimation. The specific implementation process is as follows:

Establish the mathematical model of the time-varying dynamic system as:

$$\begin{cases} x_k = f(x_{k-1}, w_k) \\ y_k = h(x_k, v_k) \end{cases} \tag{11}$$

where $f(\cdot)$ is a nonlinear function of system state and variable, w_k is system noise, and v_k is system measurement noise.

a) Initialization: Sample and generate the initial particle set from known probability density $p(x_0)$, where each particle in the set has equal weight:

$$\{x_0^i, 1/N; i = 1, 2, \cdots, N\} \tag{12}$$

b) Importance Sampling: - Randomly draw N particles from the proposal distribution function $q(x_k|x_{k-1}^i, y_k)$:

$$x_k^i \sim q(x_k|x_{k-1}^i, y_k), \quad i = 1, 2, \cdots, N \tag{13}$$

- Update particle weights and normalize them:

$$w_k^i \propto w_{k-1}^i \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, y_k)} \tag{14}$$

$$w_k^i = \frac{w_k^i}{\sum_{i=1}^N w_k^i} \tag{15}$$

c) Resampling: Calculate effective particle number $\hat{N}_{\text{eff}}$. If $\hat{N}_{\text{eff}} < N_{\text{thr}}$, perform resampling, selecting particles with larger weights w_k^i for replication and deleting those with smaller weights to form a new generated particle set $\{x_k^j, 1/N; j = 1, 2, \cdots, N\}$.

d) State Estimation: Estimate system state and variance:

$$\hat{x}_k \approx \sum_{i=1}^N w_k^i x_k^i \tag{16}$$

$$\hat{P}_k \approx \sum_{i=1}^N w_k^i (x_k^i - \hat{x}_k)(x_k^i - \hat{x}_k)^T \tag{17}$$

e) Iteration: Set $k = k + 1$ and return to step b).

2.2 Improved Resampling Particle Filter Algorithm

The traditional resampling algorithm discards small-weight particles and repeatedly copies large-weight particles during new particle set generation, which causes loss of particle diversity. To solve this problem, we introduce a distance comparison process and optimized combination strategy during resampling. First, the distance comparison process preserves medium-weight particles as much as possible. Second, according to the optimized combination strategy, new particles are derived from large-weight particles instead of repeated copying, thereby overcoming particle degeneracy while enhancing both the effectiveness and diversity of the new particle set. The specific implementation steps are as follows:

1) Distance Comparison Process: To ensure medium-weight particles are correctly retained after resampling, we introduce a distance comparison process in the mapping from cumulative density function to uniform distribution random numbers. The distance comparison calculation is defined as:

$$\|c_i - u_j\| < \delta \cdot \|c_i - c_{i-1}\| \tag{18}$$

where c_i represents the cumulative density function value of the particle with index i, u_j is the nearest random number, and δ is the distance factor. If the distance between c_i and u_j is much smaller than the particle' s own weight, the particle is considered to have medium weight and should be preserved after resampling. By introducing distance comparison at each resampling moment, medium-weight particles can be correctly output after resampling execution, enhancing the effectiveness of the particle set.

2) Generating New Particles Based on Optimized Combination Strategy: In the resampling process, particle set diversity loss is caused by excessive replication of large-weight particles. To avoid sample impoverishment, we propose generating new offspring particle sets through an optimized combination strategy.

During resampling, particles with large weights are regarded as root particles and combined with state information from neighboring particles to generate new offspring particle sets. The optimized combination formula is defined as:

$$x_m^i = \sum_{j=1}^{m} \alpha_j \cdot x_a^j, \quad i = 1, 2, \cdots, m \tag{19}$$

where x_m^i represents the new particle set generated from root particles, and x_a^j represents the neighboring particle set adjacent to the particle. The parameter α_j is defined as:

$$\alpha_j = \frac{w_a^j}{\sum_{j=1}^{m} w_a^j} \cdot \frac{1}{\|x_a^j - x_{\text{root}}\|} \tag{20}$$

where w_a^j represents the importance weight of a specific neighboring particle, and $\|x_a^j - x_{\text{root}}\|$ represents the spatial interval between this specific neighboring particle and the root particle. By introducing parameters, both the contribution of different neighboring particle weights and their spatial interval impact from the root particle are considered.

Through combination with neighboring particle sets, the generated new particle set contains more state information from neighboring particles, enhancing the diversity of the final output particle set and enabling better expression of the system' s posterior probability density.

3.1 LSTM Model Training Based on Improved Particle Filter Algorithm

The process of optimizing functions using particle filter algorithms can be considered a time-varying dynamic system. Iteration times can be regarded as discrete time values, while system states can be viewed as optimal solutions at each iteration. First, we need to determine the measurement equation, update equation, and initial state of the time-varying dynamic system.

The goal of the LSTM prediction model is to minimize the error between predicted and actual system values. That is, the objective function is to minimize $\sum_{t=1}^{T}(y_t' - y_t)^2$, where y_t' is the actual value and y_t is the system output value, with T being the length of the time series. The system measurement equation can be defined as:

$$\text{fitness} = \sum_{t=1}^{T}(y_t' - y_t)^2 \tag{21}$$

The system update equation consists of equations (3)-(8). With the system update equation, measurement equation, and initial state known, the particle filter algorithm solves the state optimal estimation problem. The basic steps for LSTM parameter optimization based on particle filter algorithm are as follows:

a) System State Initialization: Calculate LSTM output values y_t according to the LSTM network parameters and equations (3)-(8). Determine the definition domain of the initial system state based on error range as system constraint conditions. Particle filter algorithm performance is less affected by initial particle group values, so this algorithm adopts the method of randomly generating populations within its definition domain for initialization.

b) Importance Sampling: Let particle total number be N and iteration total number be t_f. For $i = 1, 2, \cdots, N$, randomly draw N particles from the initial particle set according to probability distribution function $p(x_0)$.

c) Update Global Optimal Solution: Evaluate the fitness values of all particles x_k^i according to the system measurement equation (22), selecting the optimal particle and corresponding optimal fitness value.

d) Update Particle Weights and Normalize: If particles do not satisfy system constraint conditions, set particle weights to 0; if they satisfy constraints, calculate each particle' s weight according to current observation value y_k and equation (13), then normalize particle weights through equation (14).

e) Resampling: To reduce particle degeneracy and ensure particle set effectiveness and diversity, this algorithm introduces a distance comparison process to guarantee correct output of medium-weight particles. During generation of new support particle sets, it introduces an optimized combination method that combines neighboring particle states to generate new particle sets, overcoming sample impoverishment caused by excessive replication of large-weight particles.

Calculate effective particle number $\hat{N}_{\text{eff}}$. If $\hat{N}_{\text{eff}} < N_{\text{thr}}$, execute the resampling process according to distance comparison and optimized combination strategy to obtain new support particle set x_k^i.

f) State Estimation: Estimate system state and variance according to equations (16) and (17).

g) Termination Condition Check: Determine whether the algorithm termination condition is satisfied based on iteration steps k. If not satisfied, return to step 2.

h) Algorithm Termination: Return the optimal particle and system output value y_t.

3.2 Establishment of LSTM Prediction Model Based on Particle Filter Algorithm

In network structure design, the LSTM network structure is determined to consist of 1 input layer, 1 hidden layer, and 1 output layer. Based on Figure 4 [Figure 4: see original paper], the specific prediction steps are as follows:

a) Data Preprocessing and Sample Division: Perform extreme difference standardization processing on network traffic sample data to normalize sample data within [0,1]. Let $X_{\max}$ and $X_{\min}$ represent the maximum and minimum values in the sample data. After processing, the data is divided into training and test sets using simple cross-validation, with the first 90% of data groups as the training set for LSTM network model training and the remaining 10% as the test set.

b) Determine LSTM Network Parameters: Set the number of input layer units, output layer units n, and hidden layer units l, as well as hidden layer parameters. Set network input as m variables per batch as input and n variables as output. The number of hidden layer units is determined through multiple trial experiments. In this experiment, $m = 12$ and $n = 2$, meaning we predict future 2-hour network traffic values using each batch of 12-hour network traffic data.

c) Prediction Model Training Parameter Setting: Determine parameters through the improved particle filter algorithm to make the model converge to global optimum.

d) Network Traffic Prediction.

e) Prediction Model Error Evaluation: For network model prediction results, we employ two error analysis methods to verify prediction accuracy: Root Mean Square Error (RMSE) and Mean Relative Error (MRE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{22}$$

$$\mathrm{MRE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{23}$$

where y_i is the actual value of sequence samples, $\hat{y}_i$ is the predicted value, and N is the total number of samples.

4 Simulation Results and Analysis

To analyze the prediction performance of the proposed PF-LSTM model, we selected hourly network traffic statistics from a certain region from May 1, 2016 to May 20, 2016 as the experimental object, obtaining a total of 500 samples, as shown in Figure 5 [Figure 5: see original paper]. The first 450 sample points were selected as the training set to construct the network traffic learning model, and the last 50 sample points were used as the test set to analyze model performance. We verified its performance from two aspects: prediction accuracy and algorithm convergence efficiency.

a) Prediction Accuracy: Figure 6 [Figure 6: see original paper] shows the prediction result curves of network traffic values using the traditional BPTT algorithm-trained LSTM network model and the proposed PF-LSTM network model. As seen in Figure 6, although the traditional LSTM model prediction curve can reflect network traffic variation trends, it does not fit well with the actual value curve. In contrast, the PF-LSTM model converges to the global optimum and can better track network traffic variation trends, achieving more accurate predictions.

To further compare the prediction effects of LSTM and PF-LSTM models, Table 1 calculates the average prediction errors, RMSE and MRE, for both models. The table shows that the PF-LSTM model produces smaller prediction errors and significantly outperforms the traditional LSTM model. From the MRE perspective, the traditional LSTM result is 14.72%, while the PF-LSTM model result is 6.31%, demonstrating superior prediction error.

b) Convergence Efficiency: To verify the convergence efficiency of the PF-LSTM prediction model, we conducted error convergence experiments on the training dataset, comparing it with the traditional LSTM model and the GWO-LSTM time series prediction model based on the Grey Wolf Optimizer algorithm proposed in literature [19]. In the GWO-LSTM model, the number of grey wolves was set to 200. All three models had the same LSTM structure and 400 iterations, trained on the same dataset. The experimental results are shown in Figure 7 [Figure 7: see original paper]. Analyzing the final mean relative error, both PF-LSTM and GWO-LSTM models have lower steady-state relative errors than the LSTM model, indicating superior prediction accuracy. Considering convergence speed, the PF-LSTM model has the fastest error convergence speed, with initial errors significantly smaller than other models. After training on approximately 200 sets of data, it converges to the optimal error value, while GWO-LSTM and LSTM models require approximately 450 and 300 sets of data respectively before their relative error curves stabilize. Therefore, the PF-LSTM model can converge to smaller steady-state error values with the fastest convergence speed and highest efficiency.

5 Conclusion

For complex and volatile network traffic time-series data, LSTM networks are employed for time series prediction. However, the BPTT algorithm used during training tends to converge to local optimal values, resulting in unsatisfactory prediction performance. This paper proposes a short-term network traffic prediction model (PF-LSTM) based on an improved Particle Filter constrained algorithm (PF) to optimize the LSTM neural network. The model utilizes distance comparison processes and optimized combination strategies to improve the particle filter algorithm, which is then used to train the LSTM network and optimize model parameters for short-term network traffic prediction. Simulation experiments demonstrate that the PF-LSTM prediction model achieves good prediction accuracy, can accurately grasp network traffic variation trends, and exhibits excellent convergence efficiency. Applying this model to network traffic management and planning can help reduce the frequency of network traffic congestion and improve network resource utilization.

References

[1] Chen Guangju, Liang Peng, Wang Kun. Research on network traffic prediction model [J]. Information Communication, 2017, 31(8): 191-194.

[2] Chen Huafeng, Liu Jianing. Modeling and forecast of wireless network traffic based on combinatorial optimization theory [J]. Modern Electronics Technique, 2016, 39(23): 43-46.

[3] Aldhyani T H H, Joshi M R. Integration of time series models with soft clustering to enhance network traffic forecasting [C]//Proc of International Conference on Research in Computational Intelligence & Communication Networks. 2017: 212-214.

[4] Haviluddin, Alfred R. Performance of modeling time series using nonlinear autoregressive with eXogenous input (NARX) in the network traffic forecasting [C]//Proc of International Conference on Science in Information Technology. 2016: 164-168.

[5] Xu Aijun. Network traffic prediction model based on improved ABC algorithm for optimizing LSSVM [J]. Journal of Computer Applications and Software, 2015, 32(1): 323-326.

[6] Zhang Lifang, Zhang Xiping. Optimization of BP Neural Network Based on Quantum Genetic Algorithm for Network Traffic Prediction [J]. Computer Engineering and Science, 2016, 38(1): 114-119.

[7] Zhao Zheng, Chen Weihai, Wu Xingming, et al. LSTM network: a deep learning approach for short-term traffic forecast [J]. IET Intelligent Transport Systems, 2017, 11(2): 68-75.

[8] Huang Bin, Lu Jinjin, Wang Jianhua, et al. Object recognition algorithm based on deep convolutional neural network [J]. Journal of Computer Applications, 2016, 36(12): 3333-3340.

[9] Yang Weiyue, Fu Qian, Wan Dingsheng. Time Series Prediction Model Based on Deep Cyclic Neural Network [J]. Computer Technology and Development, 2017, 27(3): 35-38.

[10] Kong Weicong, Dong Zhaoyang, Jia Youwei, et al. Short-term residential load forecasting based on LSTM recurrent neural network [J]. IEEE Trans on Smart Grid, 2017, PP(99): 1.

[11] Chen Kai, Yan Zhijie, Huo Qiang. A context-sensitive-chunk BPTT approach to training deep LSTM//BLSTM recurrent neural networks for offline handwriting recognition [C]//Proc of International Conference on Document Analysis and Recognition. 2016: 411-415.

[12] Cao Wei. Improved particle filter algorithm and its application [D]. Beijing: Graduate School of Chinese Academy of Sciences, 2012.

[13] Ergen T, Kozat S S. Efficient Online Learning Algorithms Based on LSTM Neural Networks [J]. IEEE Trans on Neural Network Learn System, 2017, PP(99): 1-12.

[14] Zhang Lingxiao, Liu Kecheng, Yang Xinfeng, et al. Nonlinear constrained optimization algorithm based on improved particle filter [J]. Journal of Computer Applications, 2014, 31(11): 3266-3268.

[15] Jouin M, Gouriveau R, Hissel D, et al. Particle filter-based prognostics: Review, discussion and perspectives [J]. Mechanical Systems & Signal Processing, 2016, 31(2): 72-73.

[16] Zhou Tian, Peng Dongdong, Xu Chao, et al. An adaptive particle filter based on Kullback-Leibler distance for underwater terrain aided navigation with multi-beam sonar [J]. IET Radar Sonar Navigation, 2018, 12(4): 441-443.

[17] Chen Dawei. Research on traffic flow prediction in the big data environment based on the improved RBF neural network [J]. IEEE Trans on Industrial Informations, 2017, 13(4): 2000-2008.

[18] Li Xiaoting, Shi Jianfang. Improved particle filter algorithm based on resampling technique [J]. Microelectronics & Computer, 2016, 33(9): 164-168.

[19] Wang Shuqin, Hua Gang, Hao Guosheng, et al. Application of long-short-term memory network based on grey wolf optimization algorithm in time series prediction [J]. Chinese Scientific Papers, 2017, 16(20): 2309-2314.

[20] Lin Hunlong, Li Sai. Remaining Useful Life Prediction of the Lithium-ion Batteries Based on Particle Filter Algorithm [J]. Science Technology & Engineering, 2017, 17(29): 296-301.

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