IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 9, 2019, accepted December 27, 2019, date of publication January 3, 2020, date of current version January 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963784

PSO-ELM: A Hybrid Learning Model for Short-Term Traffic Flow Forecasting

WEIHONG CAI[1,2,3], JUNJIE YANG[1], YIDAN YU[1], YOUYI SONG[4], TENG ZHOU[1,2,4], AND JING QIN[4]

[1]Department of Computer Science, College of Engineering, Shantou University, Shantou 515063, China
[2]Key Laboratory of Intelligent Manufacturing Technology, Ministry of Education, Shantou University, Shantou 515063, China
[3]Big Data Research Institute, Tongxing Technology Development Corporation, Shantou 515000, China
[4]Center for Smart Health, School of Nursing, The Hong Kong Polytechnic University, Hong Kong

Corresponding author: Teng Zhou (zhouteng@stu.edu.cn)

This work was supported in part by the Science and Technology Planning Project of Guangdong Province under Grant 2019B010116001 and Grant 2016B010124012, in part by the NSFC under Grant 61902232 and Grant 61902231, in part by the Natural Science Foundation of Guangdong Province under Grant 2018A030313291, in part by the Education Science Planning Project of Guangdong Province under Grant 2018GXJK048, and in part by the STU Scientfic Research Foundation for Talents under Grant NTF18006.

ABSTRACT Accurate and reliable traffic flow forecasting is of importance for urban planning and mitigation of traffic congestion, and it is also the basis for the deployment of intelligent traffic management systems. However, constructing a reasonable and robust forecasting model is a challenging task due to the uncertainties and nonlinear characteristics of traffic flow. Aiming at the nonlinear relationship affecting traffic flow forecasting effect, a PSO-ELM model based on particle swarm optimization is proposed for short-term traffic flow forecasting, which takes the advantages of particle swarm optimization to search global optimal solution and extreme learning machine to fast deal with the nonlinear relationship. The proposed model improves the accuracy of traffic flow forecasting. The traffic flow data from highways A1, A2, A4, A8 connecting to Amsterdam's ring road are employed for the case study. The RMSEs of PSO-ELM model are respectively 252.61, 173.75, 200.24, 146.05, while the MAPEs of PSO-ELM model are respectively 11.86%, 10.10%, 10.74%, 11.60%. The experimental results show that the performance of the proposal is significantly better than the performance of state-of-the-art models.

INDEX TERMS Short-term traffic flow forecasting, extreme learning machine, particle swarm optimization, time-series model.

I. INTRODUCTION

Traffic flow forecasting, especially short-term traffic flow forecasting is a pivotal aspect of the intelligent transportation system, because traffic prediction is an important enabler for traffic management or traffic control systems whose aim is reducing congestion. For the drivers, they incur a longer travelling time and economical loses because of the traffic congestion, which is one of the severe problems in urban areas. Traffic flow does not only show random behaviours, which are influenced by exogenous factors, such as unexpected events or weather, but also reveals seasonality obscured by noise [1]. Due to the uncertainty and randomness of traffic flow, it is still a challenging task to construct a reasonable and robust forecasting model [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Rongbo Zhu.

Different methods and theories were proposed for traffic flow forecasting [3], which can be usually classified into parametric methods and non-parametric ones [4]. Parametric methods include moving average [5], exponential smoothing (ES) [6], auto-regressive integrated moving average (ARIMA) models [7], [8], Kalman filtering methods [9]–[11], multivariate time series models [12], [13], and spectral analysis [14]. Support vector machine regression (SVR) [15], [16], non-parameter regression models [17], artificial neural network (ANN) [18], fuzzy logic system methods [19], [20], deep feature fusion model [21] and deep belief network [22] belong to the non-parametric ones.

With the fast developing of machine learning and deep learning methods, traffic flow forecasting models based on neural network attract more and more attention and interesting. Neural networks have great advantages, such as parallel computation, flexibility and the ability to learn and build

models of nonlinear complex relationships [23], [24]. The feedforward neural networks have been widely introduced in a series of areas including traffic flow forecasting. However, they still have some drawbacks, for example, the slower learning speed of than required [25], [26] and easily converging to local minima due to gradient descent-based learning methods with improper learning steps [27]. The pivotal cause may be that all the parameters of the networks are adjusted iteratively by using slow gradient-based learning algorithms [23], [28]. In order to adjust the weight parameters of machine learning model more accurately and quickly, some data-driven machine learning (ML)-based methods apply heuristic algorithms for ML model training in the domain of forecasting. In [29], the ANN in each stage of the forecasting model employs the GA optimization technique to optimally tune the weight parameters between neurons.

Huang *et al.* [30] proposed a novel learning algorithm called extreme learning machine (ELM) and proved that the input weights and hidden layer biases can be stochastic chosen if the activation functions in the hidden layer are infinitely differentiable. In ELM, input weights and hidden layer biases are given by random initialization, and output weights matrices are calculated using Moore-Penrose (MP) generalized inverse [23]. ELM is not only thousands of times faster than traditional learning algorithms, but also averts from some problems caused by gradient-based learning methods such as local minima, stopping criteria and learning rate [31], [32]. However, the over-fitting problem still cannot be solved completely in ELM. To improve prediction accuracy, the most common method is to increase the number of hidden layer nodes. The situation of over-fitting in ELM becomes more serious when the number of hidden layer nodes become large [33].

In this study, we reformulate the extreme learning machine optimized by particle swarm algorithm for traffic flow forecasting. Our particle swarm optimized extreme learning machine keeps the competitive forecasting accuracy as well as reducing the network complexity and avoiding over-fitting. We summarize the major contributions of this work as follows.

- First, we rethink the potential to improve the extreme learning machine by optimizing the weights and biases with particle swarm optimization.
- Second, we applied our particle swarm optimized extreme learning machine that preserves its virtues to learn the nonlinear traffic flow in a end-to-end mechanism.
- Third, we demonstrate the outperformance of our PSO-ELM on two benchmark datasets by comparing with several state-of-the-art methods for traffic flow forecasting.
- Fourth, we also extend our PSO-ELM to applications of other domains, such as CO_2 emissions forecasting, and the results demonstrate the generalization of our model.

The rest of this paper is organized as follows. The second part is the related work, the third part is the methodology, and the fourth part is an empirical study from real-world data. The fifth part is the conclusions.

II. METHODOLOGY

In this section, we first formulate the traffic flow forecasting model by extreme learning machine. Then, we incorporate the particle swarm algorithm to optimize the model.

A. EXTREME LEARNING MACHINE

In the training process, the traditional feedforward neural network propagates the error back through gradient descent algorithm to constantly revise the model parameters, such as weight and threshold [34]. In this way, the sum of the squared errors increasingly reduces a certain level, and the output of the neural network gradually approaches the prospective output. Different from the traditional feedforward neural network, the extreme learning machine is a machine learning algorithm for a single-layer feedforward neural network, in which, only the output weight is needed to be calculated according to the parameters of the hidden layer, and the hidden layer is randomly set [35]. This algorithm not only has higher generalization performance and faster learning speed than the traditional gradient-based learning versions, but also has strong fitting ability, and lower computational complexity. A three layers ELM is illustrated in Fig. 1.

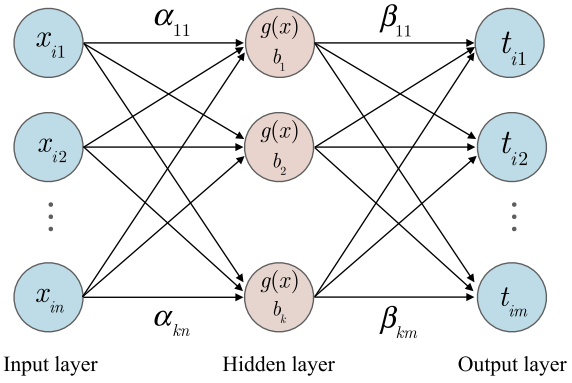

FIGURE 1. Extreme learning machine. The feedforward neural network has only one hidden layer, whose parameters include input weights α, output weights β, and hidden layer biases b.

The principle of the extreme learning machine is as follows. Assume that there are N training samples $\{(x_i, t_i)\}_{i=1}^{N}$, where $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^\top \in \mathbb{R}^n$, $t_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^\top \in \mathbb{R}^m$. The neural network with an activation function $g(x)$ and k hidden nodes are mathematically shown as

$$H\beta = T, \tag{1}$$

where $H = \{h_{ij}\}_{i=1,\ldots,N,j=1,\ldots,k}$, represents the output matrix of hidden layer. $h_{ij} = g\left(\alpha_j^\top x_i + b_j\right)$ is the output of jth hidden node in respect of x_i. $\alpha_j = [\alpha_{j1}, \alpha_{j2}, \ldots, \alpha_{jn}]^\top$ is the weight vector that links the input nodes to the jth hidden neuron. b_j is the bias of the jth hidden neuron. The matrix

of output weights is represented as $\beta = [\beta_1, \beta_2, \ldots, \beta_k]^\top$, where $\beta_j = [\beta_{j1}, \beta_{j2}, \ldots, \beta_{jm}]^\top, j = 1, \ldots, k$ is the weight vector that links the output nodes to the jth hidden neuron. $T = [t_1, t_2, \ldots, t_N]^\top$ represents the matrix of targets.

The idea of ELM is to generate the initial input weights and hidden biases randomly. Then the matrix H is determined according to the activation function $g(x)$. In this way, the training of feedforward neural network can be transformed into a problem of solving the least-squares (LS) solution to the linear system given by the Eq. 1, whose result can be denoted using Eq. 2.

$$\hat{\beta} = H^\dagger T, \tag{2}$$

where $H^\dagger$ means the Moore-Penrose (MP) generalized inverse of matrix H.

B. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is a iteration-based optimization algorithm based on the observation of the social behavior of biological organisms, such as birds in a flock. The particle swarm optimization algorithm uses the individuals sharing of information in the group to make the movement of the whole group to evolve from disorder to order in the problem solving space, so as to obtain the optimal solution [36].

Suppose that there is a population $Y = (Y_1, Y_2, \ldots, Y_n)$ consisting of n particles in a D-dimensional space, where the ith particle is represented as a D-dimensional vector $Y_i = [y_{i1}, y_{i2}, \ldots, y_{iD}]^\top$, which not only represents the position of the ith particle in the D-dimensional search space, but also represents a potential solution of the problem. According to the objective function, the fitness value corresponding to the particle position Y_i can be calculated to determine whether the current position is good or bad. Each particle has a velocity indicating the distance and the orientation. The velocity of the ith particle is $V_i = [v_{i1}, v_{i2}, \ldots, v_{iD}]^\top$. At every iteration of the PSO, the particles update their velocity and position calculated as follows.

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (p_{id}^k - y_{id}^k) + c_2 r_2 (p_{gd}^k - y_{id}^k), \tag{3}$$

$$y_{id}^{k+1} = y_{id}^k + v_{id}^{k+1}, \tag{4}$$

where k is the number of hidden layer nodes, ω is the inertia weight. $P_i = [p_{i1}, p_{i2}, \ldots, p_{iD}]^\top$ is the personal best position of the ith particle and $P_g = [p_{g1}, p_{g2}, \ldots, p_{gD}]^\top$ is the global best position of the swarm. The learning factors are denoted using terms c_1 and c_2. The terms r_1 and r_2 are randomly given in the range $U(0, 1)$. $[v_{min}, v_{max}]$ is the range of the velocity.

C. PSO-ELM FOR TRAFFIC FLOW FORECASTING

In the process of random assignment, there may be some input weight matrices and the hidden layer biases are 0, that is, a part of the hidden layer nodes is invalid. Thus, in some practical applications, lots of hidden layer nodes are needed to achieve the prospective accuracy in ELM. ELM has insufficient generalization ability in dealing with the samples that do not appear in the training process [37].

Algorithm 1 PSO-ELM Algorithm

```
Establish ELM network;
Set the number of hidden neurons k, activation function g(x);
Initialize the particles Θ;
Determine particle size n, inertia weight ω, the maximum of iterations T, and acceleration coefficients c1 and c2;
Input the training samples;
while ending conditions false do
    for all particle i of the population do
        Calculate the fitness f(Θ^i);
        Get the personal extremum P_i;
    end for
    Calculate the global extremum P_g;
    for all particle i of the population do
        Adjust the velocity V_i;
        Update the position Θ^i;
    end for
end while
Separate the global extremum P_g;
Get input weight α_j and hidden layer bias b_j;
```

Aiming at the above problems, this paper proposes an algorithm called PSO-ELM, which incorporates the particle swarm optimization and the extreme learning machine. In this model, the PSO algorithm optimizes the input weight matrix and the hidden layer bias in ELM to obtain an optimal network [36].

The number of particles is generally set from 20 to 40. The particles in the population are composed of the input weight matrices and the hidden layer biases. The particle length is represented using $D = k(n + 1)$, where k is the number of hidden layer nodes, and n is the number of input layer neurons, that is, the dimension of the input vector. $\Theta^i = [\alpha_{11}^i, \alpha_{12}^i, \ldots, \alpha_{1n}^i, \alpha_{21}^i, \alpha_{22}^i, \ldots, \alpha_{2n}^i, \ldots, \alpha_{k1}^i, \alpha_{k2}^i, \ldots, \alpha_{kn}^i, b_1^i, b_2^i, \ldots, b_k^i]$ is the ith particle in the population, where α_{ij}^i and b_j^i are random numbers in the range denoted by $[-Z_{max}, Z_{max}]$. Generally, $Z_{max} = 1$ as [36]. Choose the mean square error rate in the training process to construct the fitness function $f\left(\Theta^i\right)$. The iteration does not stop until the number of iterations exceeds the maximum or the fitness value is less than the minimum.

Suppose that we will predict the traffic flow of a point of interest (POI), we collect the traffic flow of the POI through the roadside units (RSU), which transmit real-time measurements to the intelligent transportation control center [15]. Then, we aggregate the historical traffic flow dataset $\{(x_\tau, t_\tau)\}_{\tau=1}^N$ of the POI for training, where $x_\tau = \{vol_{\tau-1}, \ldots, vol_{\tau-L}\}$, $t_\tau = vol_\tau$, and L is the number of associated time intervals. After trained our PSO-ELM, we can predict the traffic flow $\hat{vol}_{\tau+1}$ at time interval $\tau + 1$ by inputting the near traffic flow $\{vol_\tau, \ldots, vol_{\tau-L+1}\}$. Without loss generality, our algorithm can be extend to co-predict the traffic flow of multi-POI.

The training steps of our PSO-ELM are detailed in Algorithm 1.

III. CASE STUDY

In this section, the traffic flow data from four highways A1, A2, A4, A8 connecting to Amsterdam's ring road (A10 motorway) are employed for case study.

A. DATA DESCRIPTION

The real world data come from four highways including A1, A2, A4 and A8. As shown in Fig. 2, these four roads end on Amsterdam's ring road (A10 highway) [38].

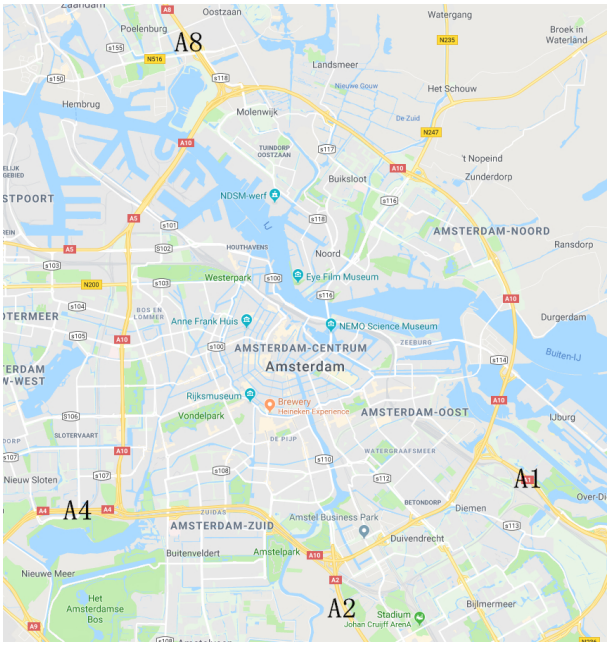

FIGURE 2. The four motorways, named A1, A2, A4, and A8, end on the ring road of Amsterdam. Sensors are placed on each POI to collect the traffic flow data.

We briefly describe the four highways as follows. All four highways meet the ring road A10. Inducted loops are placed close to the junctions on each motor road. The data were available from May 20, 2010 to June 24, 2010 in 1-min aggregation, which are the number of vehicles per hour collected by the sensor.

The erroneous data that are zero or negative for a long time is mixed in the raw data. We use the average of the measurements at the same time in other weeks to make a simple correction to the erroneous data.

B. EVALUATION CRITERIA

The experiment used two commonly employed criteria to assess the performance of our PSO-ELM. The average differences between the measurements and the predictive values of the method are measured by the root mean square error (RMSE). The mean absolute percentage error (MAPE) denotes the percentage of the differences. The two criteria are calculated using Eq.5 and 6, respectively.

$$RMSE = \sqrt{\frac{1}{M}\sum_{m=1}^{M}[\hat{v}(m) - v(m)]^2}, \tag{5}$$

$$MAPE = \frac{1}{M}\sum_{m=1}^{M}\left|\frac{\hat{v}(m) - v(m)}{v(m)}\right| \times 100\%, \tag{6}$$

where $v(m)$ and $\hat{v}(m)$ are the true measurement and the predictive value of the mth group of data.

C. EXPERIMENTAL SETUP

As discussed in [2], [4], [38], the focus of the traffic flow forecasting is not to predict minute-by-minute undulations. Therefore, we calculate the 10-minute average based on the 1-minute aggregation in subsequent 10 min as the original data for the forecasting task.

We divide the original data into two parts. The first part consists the data in the first four weeks, which is used to train the model. The other part contains the data of last week, which is used for testing. Each input vector of the prediction model consists of 10 consecutive data values in the original data, and the corresponding output value is the forecasting value of the 11th data after the ten data. Therefore, the dimension of the network is 10, and the dimension of the output vector is 1. In addition, we set 100 neuron nodes for the hidden layer in the ELM, and set the range of inertia weights in PSO as [0.30, 0.90] according to the conventional practice. We set the number of particles to 40, and the number of iterations to 100. According to the above parameter settings, the PSO-ELM model is constructed for experiments.

We manually adjust the parameter value that denotes the input vector dimension of the ELM model mentioned above, and compare the MAPEs of the prediction results in different values, as shown in Fig. 3. Finally, we choose 10-dimensional input vector for the experiment. For the number of iterations in PSO, we set it to 100 in advance. Taking the experiments of the four data sets, the effects of iterations are shown in Fig. 4. It can be seen that the value of the fitness function tends to be stable when the number of iterations exceeds 80. So the number of iterations we set for this experiment is a reasonable parameter value.

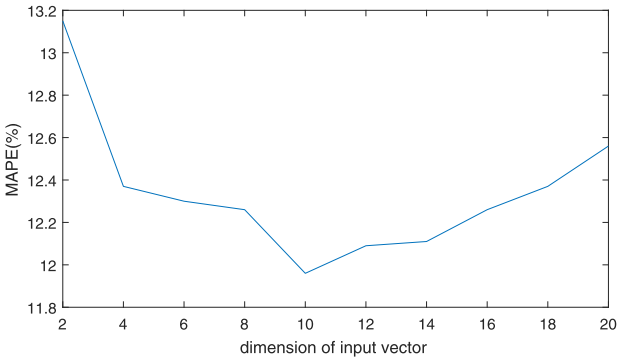

FIGURE 3. The MAPEs of the prediction results by different dimension of input vector on A1 dataset. When the dimension of input vector is 10, the prediction result achieves the best MAPE.

D. PERFORMANCE EVALUATION

Table 1 shows the comparing results of the proposed method and several models commonly applied for traffic flow forecasting.

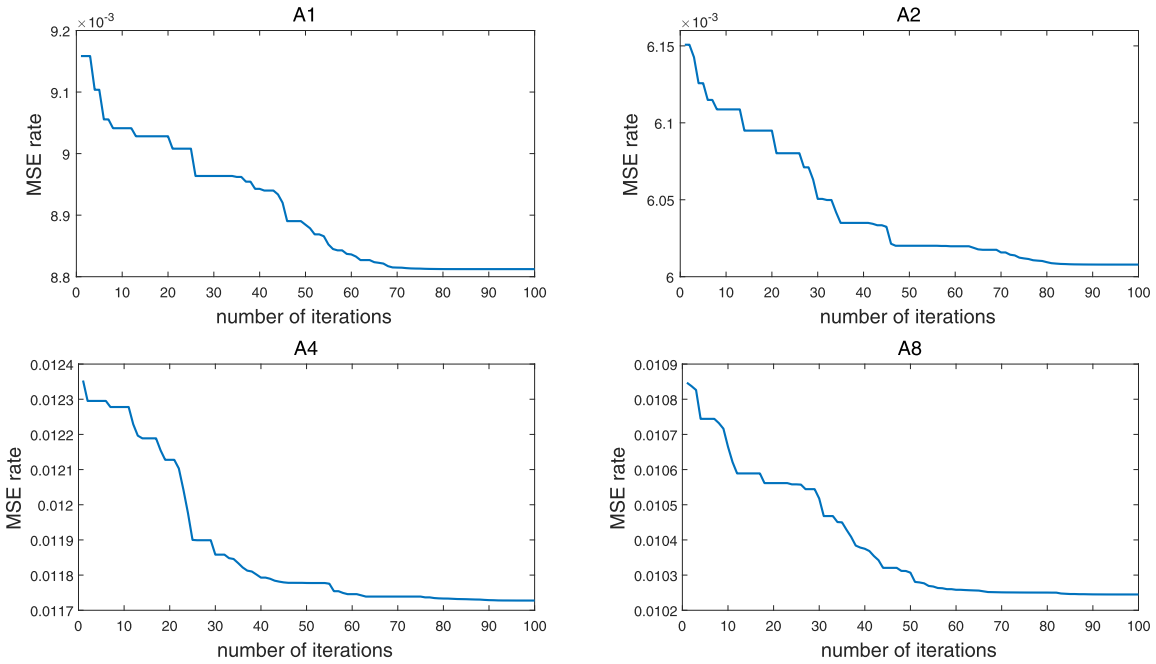

FIGURE 4. The prediction results by different number of iterations of PSO on dataset A1, A2, A4 and A8. When the number of iterations exceeds 80, the value of the fitness function tends to be stable, which illustrates it is reasonable to set the number of iterations to 100.

TABLE 1. The forecasting results of PSO-ELM and other contrast models on the dataset A1, A2, A4 and A8.

		A1	A2	A4	A8
HA	RMSE	404.84	348.96	357.85	218.72
	MAPE	16.87	15.53	16.72	16.24
SARIMA	RMSE	308.44	221.08	228.36	169.36
	MAPE	12.81	11.25	12.05	12.44
SVR	RMSE	329.09	259.74	253.66	190.30
	MAPE	14.34	12.22	12.23	12.48
ES	RMSE	315.82	226.40	237.76	174.67
	MAPE	11.94	10.75	11.97	12.00
GM	RMSE	347.94	261.36	275.35	189.57
	MAPE	12.49	10.90	13.22	12.89
ANN	RMSE	299.64	212.95	225.86	166.50
	MAPE	12.61	10.89	12.49	12.53
KF	RMSE	332.03	239.87	250.51	187.48
	MAPE	12.46	10.72	12.62	12.63
ELM	RMSE	282.99	197.44	223.72	171.98
	MAPE	11.96	10.25	11.85	13.47
PSO-ELM	RMSE	252.61	173.75	200.24	146.05
	MAPE	11.86	10.10	10.74	11.60

Historical average (HA) predicts for a given time of the day the average of the same time in the same day in previous weeks. The seasonal autoregressive integrated moving average (SARIMA) model building process is designed to take advantage of the association in the sequentially lagged relationships that usually exists in data collected periodically. The SARIMA model incorporates both non-seasonal and seasonal factors in a multiplicative model. The SARIMA model family is generally denoted as $SARIMA(p, d, q) \times (P, D, Q)_S$, where p is the number of time lags of the autoregressive model, d is the number of times the data have had past values subtracted, q is the order of the moving-average model, S refers to the number of periods in each season, and the uppercase P, D, Q refer to the autoregressive, differencing, and moving average terms for the seasonal part of the model [39]. In this study, we employ $SARIMA(1, 0, 1) \times (0, 1, 1)_{1,008}$ with $\phi = 0.8$, $\theta = 0.4$, and $\Theta = 0.8$ for such forecasting task. Comparison studies of the historical average and the SARIMA have been reported in [40].

The hybrid particle swarm optimization support vector regression (SVR) is described in detail in [41]. The radial basis function (RBF) is selected as the kernel. The width parameter γ of the RBF is optimized using particle swarm optimization. The cost parameter C is set based on the difference between the traffic flows. The PSO also optimizes the ε-insensitive loss for the SVR.

The exponential smoothing algorithm (ES) is reasonably used in [6]. ES is a special weighted moving average method (MA). The weights given by the observations at different times are not equal, which further strengthens the effect of the recent observations on the predicted values during the observation period. In this study, we used the double exponential smoothing method and set the parameter alpha in the model to 0.4, which reflects the smoothness of the trend change.

The grey prediction model (GM) shows a good application effect in [42]. GM finds the law of change by generating raw data. Here, the cumulative generation process and the $GM(1, 1)$ model are used to forecast the traffic flow, which means the closer data influences the results more.

Artificial neural network (ANN) is a learning model that is generated by the interconnection of neurons. We set the network parameters of the model according to the criteria in [18], where the number of hidden layers is set as 1,

the mean squared error goal is set as 0.001, and the spread of radial basis functions is set as 2000. Simultaneously, we set the maximum number of neurons in the hidden layer (MN) to 40. The number of neurons to add between displays (DF) is set based on default value that is 25.

Kalman filtering (KF) for traffic flow forecasting is described in detail in [43]. We set the variance of the process error Q to $0.1 \times I$, where I denotes the identity matrix. The measurement is regarded to be correct, so we set the variance of the measurement noise as 0. We set the initial state to $[\frac{1}{n}, \ldots, \frac{1}{n}]$, where n is set as 8, as suggested in [43]. $10^{-2} \times I$ is used to represent the covariance matrix of initial state estimation error.

We also conduct the experiments by standard extreme learning machines. The last one is our proposed PSO-ELM model.

We find in Table 1 that ANN has the best performance among several other methods. The MAPEs of PSO-ELM model are obviously better than the MAPEs of ANN. The RMSEs of PSO-ELM model are 15.70%, 18.41%, 11.34%, and 12.28% lower than the RMSEs of ANN at A1, A2, A4, and A8, respectively. Comparing the predictions of standard ELM and PSO-ELM, the MAPEs of PSO-ELM model in each case are better than those of the standard ELM, while

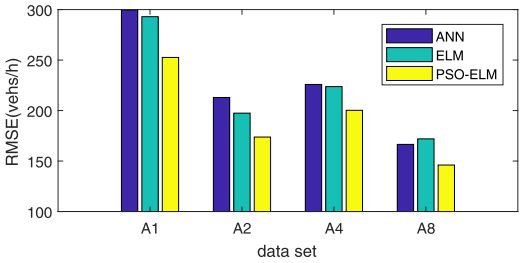

FIGURE 5. The comparisons of prediction results in different models for datasets A1, A2, A4, A8. ANN that has the best forecasting performance among several other methods and the basic ELM are chosen to compare with PSO-ELM by RMSE, which demonstrates that PSO-ELM outperforms other in RMSEs.

the RMSEs of PSO-ELM are reduced by 13.78%, 11.99%, 10.50%, and 15.08%, respectively, as shown in Fig. 5. It can be seen that the optimization of PSO has significantly improved the prediction performance of ELM.

Then, several predicting scenarios are reported to demonstrate the effectiveness of PSO-ELM in coping with uncertainties and variations of traffic flow. The measurements are plotted with red lines, while the predictions of PSO-ELM are drawn with lines that are green in Fig. 6. The related

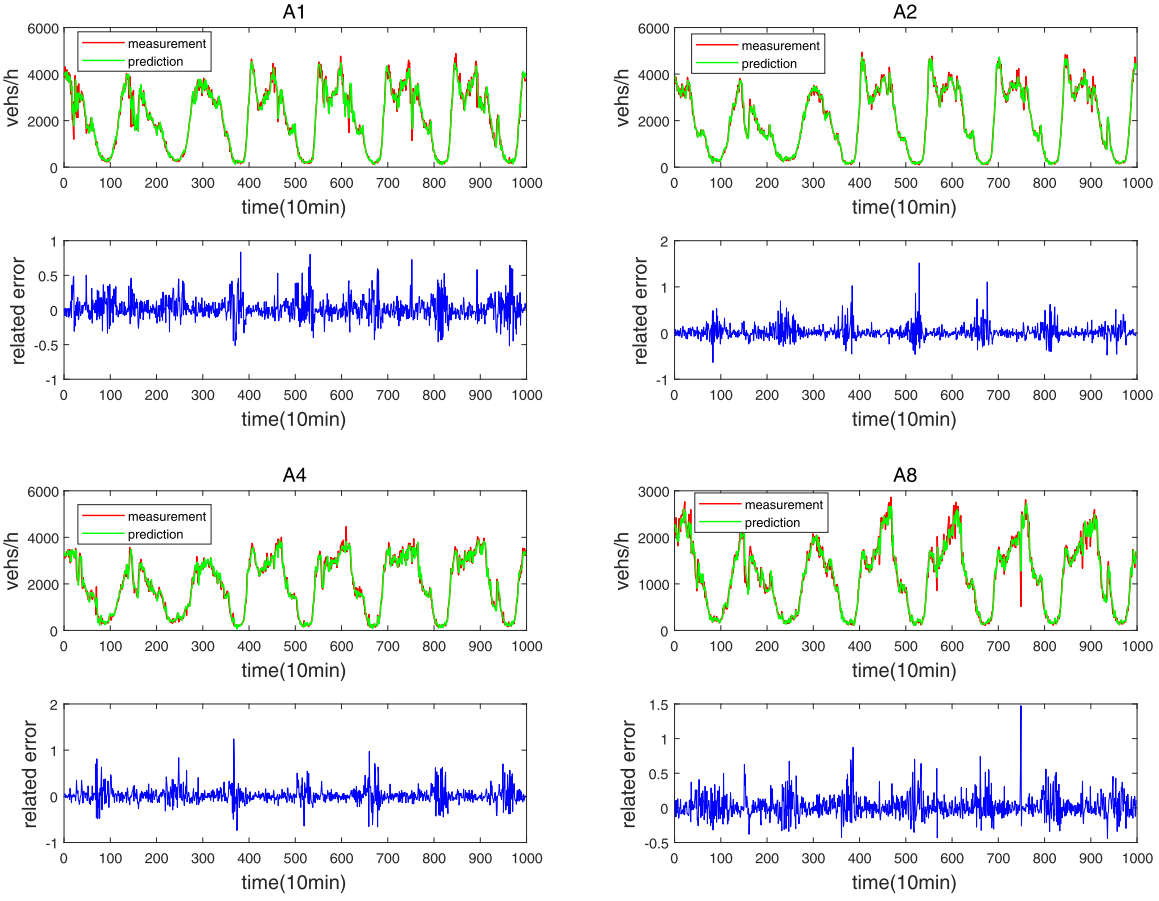

FIGURE 6. The predictions of the PSO-ELM and the measurements in a week, and the prediction related error, respectively. For these cases, where the traffic flow is very low in the morning or late at night, small prediction error still causes a large related error.

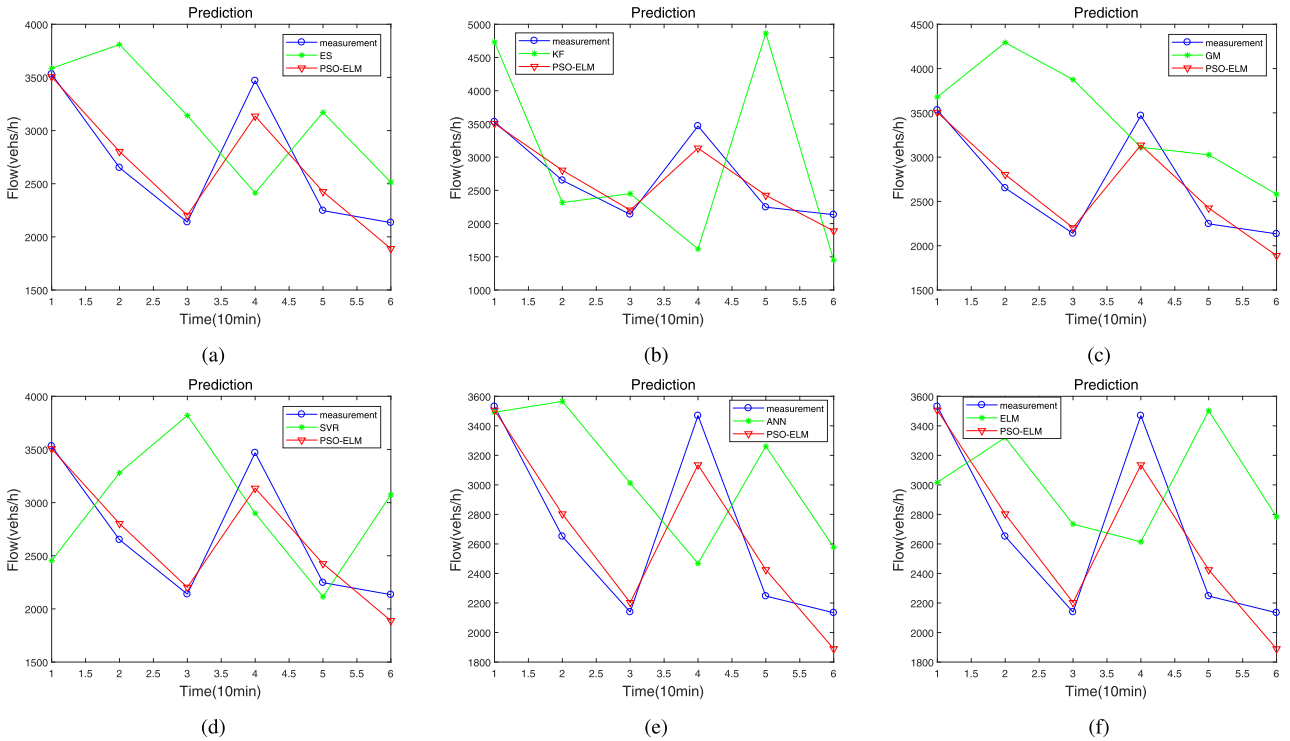

FIGURE 7. Fig.7 shows the predictions of various methods under the typical scenario where the traffic flow changes quite greatly. It is illustrated that PSO-ELM implements higher prediction accuracy in dealing with uncertainties and variations of the traffic flow.

error represents the error between the predicted and measured values divided by the measurements, plotted with blue lines. As shown in Fig. 6, except for the traffic flow that is very low in the late night or in the early morning, the PSO-ELM accomplishes gratifying high accuracy most of the time. For these cases, although prediction error is small, the related error is still very large. Fortunately, we are more concerned with forecasting accuracy when the traffic load is heavy.

At last, we illustrate that PSO-ELM outperforms on the effectiveness in coping with uncertainties and variations of the traffic flow through reporting the typical scenario shown in Fig. 7$a-f$. The other models are difficult to implement satisfactory result for forecasting in the scenario where the traffic flow changes quite greatly.

ES and KF are incapable of attaining precise results when the traffic flow has nonlinear variations. This is because the functions forms of the parametric models are too stiff to fit. The case we outcrop in Fig. 7a, b shows this problem. Especially, in Fig. 7b, the Kalman filter is overshooting on the 1st 10-min and the 5th 10-min.

Facing such a situation shown in Fig. 7c, the prediction error of GM is relatively larger for the traffic flow with obvious fluctuations. The reason is that GM generates sequences by accumulating generation operator (AGO), which can filter out some irregular changes. Thus, this determines that GM is insensitive to the changes in volatility.

Fig. 7d compares the predictions of the SVR model and PSO-ELM model in the case. It is obviously that the predictions of SVR model diverge far from the measurements.

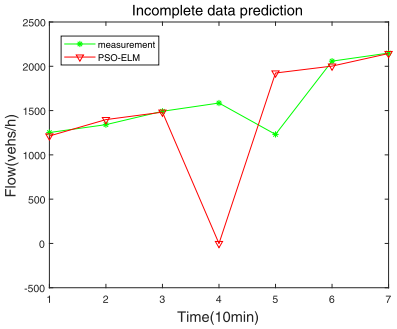

FIGURE 8. The predictions of PSO-ELM model under the scenario where the measurements include outlier. It is illustrated that PSO-ELM keeps stable predictive effect on the incomplete data.

The reason is that the input is mapped into a higher dimensional space in the SVR model for learning a function between the output and the input. The SVR model is trained in a supervised manner, and it is difficult to be trained effectively within scant training data set.

Fig. 7e, f show the measurements and predictions of ANN and the basic ELM. Neural networks have great advantages such as the ability to learn and build models of nonlinear complex relationships and flexibility. But in this case where the traffic flow varies greatly, the better performance of the proposed model can be saw in Fig. 7e, f. Certainly, PSO brings better fitting ability to PSO-ELM.

We choose a short period of time when the data are incomplete caused by hardware failure to evaluate our PSO-ELM. Fig. 8 shows the incomplete measurement causes a minor

TABLE 2. The forecasting results of PSO-ELM and several other models on the data set 1108380, 1108439, 1108599, 1111565, 1114254 and 1114515. The PSO-ELM model has clearly achieved better performance than other contrast models for short-term traffic flow forecasting.

		1108380	1108439	1108599	1111565	1114254	1114515
SVR	RMSE	22.76	36.88	22.89	26.76	35.74	17.45
	MAPE	11.72	7.77	10.73	11.86	9.50	0.1382
ES	RMSE	26.34	43.37	27.05	28.74	43.76	34.02
	MAPE	10.94	7.74	10.42	10.26	8.99	10.01
GM	RMSE	38.15	53.72	30.66	54.59	54.59	26.19
	MAPE	11.89	8.68	12.31	9.93	9.93	13.94
ANN	RMSE	25.32	41.55	25.53	27.14	37.71	20.21
	MAPE	10.73	7.82	10.31	10.94	8.89	12.82
KF	RMSE	27.68	46.33	29.83	30.20	47.76	23.56
	MAPE	11.81	8.92	11.25	11.14	9.77	14.11
ELM	RMSE	24.71	44.90	24.34	26.21	39.93	20.19
	MAPE	10.39	7.91	9.83	9.71	8.52	13.13
PSO-ELM	RMSE	23.45	38.55	23.44	25.23	38.51	18.26
	MAPE	10.27	7.36	9.80	9.69	8.38	12.44

oscillation on the next prediction value, but the prediction returns to normal quickly.

To confirm the effectiveness of the proposal in longer scenes, we choose the benchmark data set collected from Caltrans Performance Measurement System (PeMS) as the experimental data sets to compare the MAPEs and RMSEs of the representative models. The data covering the period January 2018 to July 2018 are collected from individual detectors, which span the freeway system across all major metropolitan areas of the state of California. Table 2 shows PSO-ELM outperforms others using longer data streams.

In addition to prediction accuracy, we also record the speed performance and the memory consumption of the methods forecasting on the Amsterdam's A1 dataset. As table 3 show, though parametric methods mostly have faster speed and cost less memory, manually determining parameters often costs much time and expertise domain knowledge. For the adaptive models, ELM and PSO-ELM outperform than others on these performance, especially running speed.

TABLE 3. The the speed performance and the memory consumption of PSO-ELM and several other models on the data set *A*1. The PSO-ELM model has clearly achieved better performance than others in the adaptive models, especially running speed.

Method	Running time (Second)	Memory consumption (Kilobyte)
ES	0.003	48
KF	0.144	2760
GM	34.881	2884
ANN	10.487	20956
ELM	0.027	244
PSO-SVR	482.835	128374
PSO-ELM	116.243	145532

To demonstrate the generalization of the proposed algorithms, we also extend the algorithm to other domains, such as CO_2 forecasting. We evaluate our model on the dataset of CO_2 concentration from April 1958 to December 2001 by comparing with two other models. The grey forecasting model ($GM(1,1)$) was developed by Lu *et al.* [44] to capture trends of the number of CO_2 emissions in Taiwan during $2007-2025$. The flexible artificial neural network (FANN) model was proposed by Gallo *et al.* [45] for short-term CO_2 emission forecasting. Our PSO-ELM performs better than comparisons in Table 4. In this regard, our hybrid learning model is also suitable for other forecasting tasks.

TABLE 4. The forecasting results of the PSO-ELM, $GM(1,1)$ and FANN on the atmospheric CO_2 concentration dataset.

Method	MAPE (%)	RMSE (ppmv)
$GM(1,1)$	0.115	0.530
FANN	0.100	0.487
PSO-ELM	0.081	0.379

IV. CONCLUSION

In this paper, we develop a particle swarm optimization incorporated extreme learning machine algorithm for short-term traffic flow forecasting. The particle swarm optimization effectively improves the generalization performance in term of short-term traffic flow forecasting. Such capacity empowers the PSO-ELM better performance than other control models. Extensive experiments has proven the extreme learning machine incorporating particle swarm optimization is effective for short-term traffic flow forecasting. The extension of this algorithm to other similar application fields is experimented and discussed in this study and will be promoted in the future studies.

REFERENCES

[1] Y. Zhang, Y. Zhang, and A. Haghani, "A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 65–78, Jun. 2014.
[2] T. Zhou, G. Han, X. Xu, Z. Lin, C. Han, Y. Huang, and J. Qin, "Δ-agree AdaBoost stacked autoencoder for short-term traffic flow forecasting," *Neurocomputing*, vol. 247, pp. 31–38, Jul. 2017.
[3] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, to be published.
[4] T. Zhou, G. Han, X. Xu, C. Han, Y. Huang, and J. Qin, "A learning–based multimodel integrated framework for dynamic traffic flow forecasting," *Neural Process. Lett.*, vol. 49, no. 1, pp. 407–430, Feb. 2019.
[5] T. Mai, B. Ghosh, and S. Wilson, "Multivariate short-term traffic flow forecasting using Bayesian vector autoregressive moving average model," in *Proc. Transp. Res. Board*, vol. 2, 2012, pp. 3712–3728.

[6] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, "Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 644–654, Jun. 2012.

[7] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 50–64, Jun. 2014.

[8] G. Shi, J. Guo, W. Huang, and B. M. Williams, "Modeling seasonal heteroscedasticity in vehicular traffic condition series using a seasonal adjustment approach," *J. Transp. Eng.*, vol. 140, no. 5, May 2014, Art. no. 04014012.

[9] T. Zhou, D. Jiang, Z. Lin, G. Han, X. Xu, and J. Qin, "Hybrid dual Kalman filtering model for short-term traffic flow forecasting," *IET Intell. Transp. Syst.*, vol. 13, no. 6, pp. 1023–1032, Jun. 2019.

[10] L. Cai, Z. Zhang, J. Yang, Y. Yu, T. Zhou, and J. Qin, "A noise-immune Kalman filter for short-term traffic flow forecasting," *Phys. A, Stat. Mech. Appl.*, vol. 536, Dec. 2019, Art. no. 122601.

[11] S. Zhang, Y. Song, D. Jiang, T. Zhou, and J. Qin, "Noise-identified Kalman filter for short-term traffic flow forecasting," in *Proc. IEEE 15th Int. Conf. Mobile Ad-Hoc Sensor Netw.*, 2019, pp. 1–5.

[12] T. Ma, Z. Zhou, and B. Abdulhai, "Nonlinear multivariate time–space threshold vector error correction model for short term traffic state prediction," *Transp. Res. B, Methodol.*, vol. 76, pp. 27–47, Jun. 2015.

[13] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 4, pp. 606–616, Aug. 2011.

[14] T. T. Tchrakian, B. Basu, and M. O'mahony, "Real–time traffic flow forecasting using spectral analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 519–526, Jun. 2012.

[15] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive multi-kernel SVM with spatial–temporal correlation for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2001–2013, Jun. 2019.

[16] L. Cai, Q. Chen, W. Cai, X. Xu, T. Zhou, and J. Qin, "SVRGSA: A hybrid learning based model for short-term traffic flow forecasting," *IET Intell. Transp. Syst.*, vol. 13, no. 9, pp. 1348–1355, Sep. 2019.

[17] Z. Zheng and D. Su, "Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 143–157, Jun. 2014.

[18] J. Z. Zhu, J. X. Cao, and Y. Zhu, "Traffic volume forecasting based on radial basis function neural network with the consideration of traffic flows at the adjacent intersections," *Transp. Res. C, Emerg. Technol.*, vol. 47, pp. 139–154, Oct. 2014.

[19] O. A. Arqub and Z. Abo-Hammour, "Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm," *Inf. Sci.*, vol. 279, pp. 396–415, Sep. 2014.

[20] O. A. Arqub, M. Al-Smadi, S. Momani, and T. Hayat, "Application of reproducing kernel algorithm for solving second-order, two-point fuzzy boundary value problems," *Soft Comput.*, vol. 21, no. 23, pp. 7191–7206, Dec. 2017.

[21] L. Li, X. Qu, J. Zhang, Y. Wang, and B. Ran, "Traffic speed prediction for intelligent transportation system based on a deep feature fusion model," *J. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 605–616, Nov. 2019.

[22] L. Li, L. Qin, X. Qu, J. Zhang, Y. Wang, and B. Ran, "Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm," *Knowl.-Based Syst.*, vol. 172, pp. 1–14, May 2019.

[23] R. Ahila, V. Sadasivam, and K. Manimala, "An integrated PSO for parameter determination and feature selection of ELM and its application in classification of power system disturbances," *Appl. Soft Comput.*, vol. 32, pp. 23–37, Jul. 2015.

[24] Y. Chen, F. Guo, H. Xiang, W. Cai, and X. He, "Balanced odd–variable RSBFs with optimum AI, high nonlinearity and good behavior against FAAs," *IEICE Trans. Fundam.*, vol. 102, no. 6, pp. 818–824, Jun. 2019.

[25] F. Fernández-Navarro, C. Hervás-Martínez, J. Sanchez-Monedero, and P. A. Gutiérrez, "MELM–GRBF: A modified version of the extreme learning machine for generalized radial basis function neural networks," *Neurocomputing*, vol. 74, no. 16, pp. 2502–2510, Sep. 2011.

[26] Y. Chen, L. Zhang, D. Tang, and W. Cai, "Translation equivalence of Boolean functions expressed by primitive element," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E102.A, no. 4, pp. 672–675, Apr. 2019.

[27] Y. Chen, F. Guo, and J. Ruan, "Constructing odd-variable RSBFs with optimal algebraic immunity, good nonlinearity and good behavior against fast algebraic attacks," *Discrete Appl. Math.*, vol. 262, pp. 1–12, Jun. 2019.

[28] Y. Chen, F. Guo, Z. Gong, and W. Cai, "One note about the Tu-Deng conjecture in case w(t)=5," *IEEE Access*, vol. 7, pp. 13799–13802, 2019.

[29] D. Zheng, M. Shi, Y. Wang, A. Eseye, and J. Zhang, "Day–ahead wind power forecasting using a two–stage hybrid modeling approach based on SCADA and meteorological information, and evaluating the impact of input–data dependency on forecasting accuracy," *Energies*, vol. 10, no. 12, p. 1988, Dec. 2017.

[30] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[31] Q.-Y. Zhu, A. Qin, P. Suganthan, and G.-B. Huang, "Evolutionary extreme learning machine," *Pattern Recognit.*, vol. 38, no. 10, pp. 1759–1763, Oct. 2005.

[32] J. Cao, Z. Lin, and G.-B. Huang, "Composite function wavelet neural networks with extreme learning machine," *Neurocomputing*, vol. 73, nos. 7–9, pp. 1405–1416, Mar. 2010.

[33] Y. Cui, J. Zhai, and X. Wang, "Extreme learning machine based on cross entropy," in *Proc. IEEE Int. Conf. Mach. Learn. (ICMLC)*, vol. 2, Jul. 2016, pp. 1066–1071.

[34] M. Q. Raza and A. Khosravi, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings," *Renew. Sustain. Energy Rev.*, vol. 50, pp. 1352–1372, Oct. 2015.

[35] E. M. Figueiredo and T. B. Ludermir, "Effect of the PSO topologies on the performance of the PSO–ELM," in *Proc. Brazilian Symp. Neural Netw.*, Oct. 2012, pp. 178–183.

[36] E. M. Figueiredo and T. B. Ludermir, "Investigating the use of alternative topologies on performance of the PSO–ELM," *Neurocomputing*, vol. 127, pp. 4–12, Mar. 2014.

[37] S. F. Mahmood, M. H. Marhaban, F. Z. Rokhani, K. Samsudin, and O. A. Arigbabu, "FASTA–ELM: A fast adaptive shrinkage/thresholding algorithm for extreme learning machine and its application to gender recognition," *Neurocomputing*, vol. 219, pp. 312–322, Jan. 2017.

[38] Y. Wang, J. H. Van Schuppen, and J. Vrancken, "Prediction of traffic flow at the boundary of a motorway network," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 214–227, Feb. 2014.

[39] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. Melbourne, VIC, Australia: OTexts, 2018.

[40] M. Lippi, M. Bertini, and P. Frasconi, "Short–term traffic flow forecasting: An experimental comparison of time–series analysis and supervised learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 871–882, Jun. 2013.

[41] W. Hu, L. Yan, K. Liu, and H. Wang, "A short-term traffic flow forecasting method based on the hybrid PSO–SVR," *Neural Process. Lett.*, vol. 43, no. 1, pp. 155–172, Feb. 2016.

[42] H. Guo, X. Xiao, and F. Jeffrey, "Urban road short-term traffic flow forecasting based on the delay and nonlinear grey model," *J. Transp. Syst. Eng. Inf. Technol.*, vol. 13, no. 6, pp. 60–66, Dec. 2013.

[43] Y. Xie, Y. Zhang, and Z. Ye, "Short–term traffic volume forecasting using Kalman filter with discrete wavelet decomposition," *Comput.-Aided Civil Eng.*, vol. 22, no. 5, pp. 326–334, Jul. 2007.

[44] I. Lu, C. Lewis, and S. J. Lin, "The forecast of motor vehicle, energy demand and CO_2 emission from Taiwan's road transportation sector," *Energy Policy*, vol. 37, no. 8, pp. 2952–2961, Aug. 2009.

[45] C. Gallo, F. Conto, and M. Fiore, "A neural network model for forecasting CO_2 emission," *Agris On-Line Papers Econ. Inform.*, vol. 6, pp. 31–36, Jun. 2014.

WEIHONG CAI received the Ph.D. degree from the South China University of Technology, in 2012. He is currently the Head of the Institute for Modern Network Technology and Information Security Studies, Shantou University, a Representative of the Tenth Member Congress of China Computer Federation, and the Director General of the Shantou Computer Society. His research interests include network and communication technologies, information security and network management, and computer application systems.

JUNJIE YANG is currently pursuing the master's degree with the Department of Computer Science and Technology, College of Engineering, Shantou University, China. His research interests include traffic flow forecasting and machine learning.

YIDAN YU is currently pursuing the master's degree with the Department of Computer Science and Technology, College of Engineering, Shantou University, China. Her research interests include urban computing and machine learning.

YOUYI SONG is currently pursuing the Ph.D. degree with the School of Nursing, The Hong Kong Polytechnic University. His research interests include intelligent traffic systems, medical image segmentation, machine learning, and shape analysis.

TENG ZHOU is currently an Assistant Professor with the Department of Computer Science, Shantou University, and also serves as a Research Associate with the Center of Smart Health, The Hong Kong Polytechnic University. His research interests include intelligent transportation systems and machine learning.

JING QIN received the Ph.D. degree from the Department of Computer Science and Engineering, The Chinese University of Hong Kong. He is currently an Assistant Professor with the Centre for Smart Health, School of Nursing, The Hong Kong Polytechnic University. His research interests are intelligent traffic systems, VR-based surgical simulation, multisensory human–computer interaction, and biomechanical modeling.