Hindawi
Scientific Programming
Volume 2017, Article ID 2073680, 12 pages
https://doi.org/10.1155/2017/2073680

Research Article

Probabilistic Forecasting of Traffic Flow Using Multikernel Based Extreme Learning Machine

Yiming Xing,[1,2] Xiaojuan Ban,[1] and Chong Guo[1]

[1] *School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China*
[2] *Engineering Training Center of Shenyang Aerospace University, Shenyang 110136, China*

Correspondence should be addressed to Yiming Xing; b20120409@xs.ustb.edu.cn

Received 7 August 2016; Accepted 1 December 2016; Published 16 March 2017

Academic Editor: Wenbing Zhao

Real-time and accurate prediction of traffic flow is the key to intelligent transportation systems (ITS). However, due to the nonstationarity of traffic flow data, traditional point forecasting can hardly be accurate, so probabilistic forecasting methods are essential for quantification of the potential risks and uncertainties for traffic management. A probabilistic forecasting model of traffic flow based on a multikernel extreme learning machine (MKELM) is proposed. Moreover, the optimal output weights of MKELM are obtained by utilizing Quantum-behaved particle swarm optimization (QPSO) algorithm. To verify its effectiveness, traffic flow probabilistic prediction using QPSO-MKELM was compared with other learning methods. Experimental results show that QPSO-MKELM is more effective for practical applications. And it will help traffic managers to make right decisions.

1. Introduction

Recently, the traffic flows maintain a steady growth in both urban and rural traffic leading to pollution, accidents, and congestion. To solve the problems, the intelligent transportation systems (ITS) are developed by many countries. The effectiveness of ITS is improved by using a lot of modern information technologies. According to the prediction period of time, traffic flow prediction can be divided into long-term, mid-term, and short-term prediction. Short-term traffic flow forecasting has become one of the main research areas of ITS. Prediction of real-time and accurate traffic flow becomes extremely important for effective traffic management systems, including traffic control, traffic induction, and vehicle routing. The problem induced by the randomness, nonlinearity, and complexity of traffic flow has compelled us to search for more reliable techniques to forecast traffic flow.

A lot of short-term traffic prediction algorithms are proposed in the literature [1–7]. Conventionally, a majority of study focus on developing accurate point-prediction method structures and learning algorithms for short-term traffic flow prediction, but these methods cannot be used for the quantitative analysis of the uncertainty of the prediction. In fact, there are lots of traffic variables influencing the results of traffic flow prediction such as weather, date and time, types, and flow parameters. The aim of the traffic flow forecasting model is to utilize these variables to predict the traffic flow.

Because of the chaotic property of traffic flow, mistakes in prediction are simply inevitable. In other words, forecast with exact-point value becomes inadequate to describe the real world information. To deal with the problem, prediction intervals (PIs) are vital for quantifying the underlying risks and uncertainties. PIs are a list of ranges including upper and lower bounds where the targets will lie. On the basis of the PIs with corresponding confidence level, the quantitative uncertainties of traffic flow predictions can supply valuable information to traffic managers for good preparation for the most severe and excellent situations in advance.

2. Literature Review

The exact point-prediction methods only supply the point forecasting value, while PIs work as intervals that consist of upper and lower bounds as well as pointing out the probability of correct forecast. PIs not only indicate the prediction accuracy, but also provide a range that targets will locate

[8, 9]. Some investigations in regression problems along with interval outputs have been conducted and they are able to be sorted into two classifications based on the amount of models they utilized. Studies in the first category use only one model, whereas they bring in other approaches to acquire interval outputs. These PIs construction approaches are usually set after a point-prediction model with particular preceding assumptions. Bayesian [10], mean-variance estimation [11], and bootstrap [12] are frequently used to obtain PIs. The main disadvantage of these existing methods is the high computational requirement. The other category focuses on learning a twin model; the PIs construction method named lower upper bound estimation (LUBE) approach is brought forward in [13]. Nevertheless, traditional neural networks (NNs) applied in the LUBE approach have the issue of high computational cost and overtraining.

ELM [14] is brought forward to train single-hidden layer feedforward neural networks (SLFNs). Although ELM extensively enhances the training effectiveness, fluctuation caused by the random input and hidden layer weights influences the steadiness of ELM in the situation of identical training data as well as model parameter [15]. The method that replaces ELM hidden layer with a kernel function makes ELM avoid choosing input and hidden layer weights randomly since the computation for hidden input is conducted by kernel function. The problem in ELM caused by random input and hidden layer parameters is solved by Kernel-ELM which gains higher stability by sacrificing the training rate [15]. Moreover, one kernel function is usually used in the standard kernel learning algorithms [16]. In our last paper [17], we proposed a single kernel extreme learning machine- (KELM-) based probabilistic forecasting method of traffic flow. On the basis of multikernel learning thought [18], the composites of two kernels may combine the good characteristics of them and have better performance than any other single kernel. The polynomial kernel function and the Gaussian kernel function are mixed as a kernel function which includes both kernels' advantages in this study.

A novel probabilistic traffic flow prediction approach is brought forward based on the multikernel ELM in this paper, which is applied to construct PIs for traffic flow. Then the output weights from MKELM models are optimized by Quantum-behaved particle swarm optimization (QPSO) [19]. The proposed approach has been examined by the practical traffic flow data. Accurate prediction results have represented the good performance of the QPSO-MKELM approach for traffic flow.

The rest of this paper is organized as follows. In Section 3 the algorithms of framework of MKELM and QPSO are presented. Section 4 proposes a novel MKELM model to construct optimal PIs for traffic flow with QPSO. The experiments to verify the effectiveness of the model are carried out in Section 5. Finally, Section 6 draws the conclusion.

3. Methodology

3.1. Multiple Kernel-Based ELM. ELM, developed by Huang et al. [20], is a novel learning algorithm for SLFNs that randomly selects hidden nodes parameters as well as ascertaining the output layer parameters of SLFNs analytically.

In a specified training dataset including n samples, (x, y) is a training sample. The formula of the SLFN with N_h hidden nodes is able to be represented as

$$\sum_{i=1}^{N_h} \beta_i g\left(w_i x + b_i\right) = y. \tag{1}$$

The output equation of ELM is able to be represented as

$$\begin{aligned} y &= F_{\text{ELM}}(x) = \sum_{i=1}^{N_h} \beta_i g\left(w_i x + b_i\right) = h(x)\beta, \\ Y &= H\beta, \\ & Y = \left[y_1, \ldots, y_n\right]^T, \ H = \left[h\left(x_1\right), \ldots, h\left(x_n\right)\right]^T. \end{aligned} \tag{2}$$

According to ELM theory [14], input as well as hidden layer parameters are able to be randomly distributed as long as the activation function becomes infinitely differentiable. As to fixed input weights w_i and the hidden layer biases b_i, to train an SLFN is simply equal to figuring out a least-squares solution $\widehat{\beta}$ of the linear system $Y = H\beta$.

The smallest norm least-squares solution of the above linear system is

$$\widehat{\beta} = H^+ Y, \tag{3}$$

where H^+ represents the Moore–Penrose generalized inverse of matrix H.

With a user-defined cost coefficient C, Huang et al. [15] optimized the computation for the output weights β. Among them, when the amount of the hidden nodes is larger compared to that of train data, the result of β is

$$\beta = H^T \left(\frac{I}{C} + HH^T\right)^{-1} Y. \tag{4}$$

The hidden layer output of every sample $h(x_i)$ is able to be considered as an nonlinear mapping of samples x_i. Then

$$\begin{aligned} HH^T &= \begin{bmatrix} h\left(x_1\right) \\ \vdots \\ h\left(x_N\right) \end{bmatrix}_{N*L} \cdot \begin{bmatrix} h\left(x_1\right) \\ \vdots \\ h\left(x_N\right) \end{bmatrix}_{N*L}^T \\ &= \begin{bmatrix} h\left(x_1\right) \cdot h\left(x_1\right) & \cdots & h\left(x_1\right) \cdot h\left(x_N\right) \\ \vdots & \ddots & \vdots \\ h\left(x_N\right) \cdot h\left(x_1\right) & \cdots & h\left(x_N\right) \cdot h\left(x_N\right) \end{bmatrix}_{N*N}. \end{aligned} \tag{5}$$

When the hidden layer characteristic mapping $h(x)$ could not be unidentified, Huang et al. [15] suggested applying a kernel function. According to theory of kernel function, the kernel matrix Ω for ELM is able to be described as follows, where $K(x_i, y_i)$ is kernel function:

$$\Omega_{\text{ELM}i,j} = h\left(x_i\right) \cdot h\left(x_j\right) = K\left(x_i, x_j\right). \tag{6}$$

So (5) can be deduced:

$$HH^T = \Omega_{\mathrm{ELM}i,j} = \begin{bmatrix} K(x_1,x_1) & \cdots & K(x_1,x_N) \\ \vdots & \ddots & \vdots \\ K(x_N,x_1) & \cdots & K(x_N,x_N) \end{bmatrix} = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix}_{N*L} \cdot \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix}_{N*L}^T, \tag{7}$$

$$h(x)H^T = \begin{bmatrix} K(x,x_1) \\ \vdots \\ K(x,x_N) \end{bmatrix}. \tag{8}$$

Then the output function of Kernel-ELM can be represented as

$$y = F_{\mathrm{ELM}}(x) = h(x)\beta = h(x)H^T\left(\frac{I}{C} + HH^T\right)^{-1} Y = \begin{bmatrix} K(x,x_1) \\ \vdots \\ K(x,x_n) \end{bmatrix} \left(\frac{I}{C} + \Omega_{\mathrm{ELM}i,j}\right)^{-1} Y. \tag{9}$$

Any kernel function which meets Mercer's theorem [21] is able to be a kernel function of KELM. The kernels have their own advantages and shortcomings, which can be divided into two types, global and local kernels [16]. In local kernels, only the data that are close to or in the proximity of each other have obvious effects on kernel values. The Gaussian kernel function is an example of a typical local kernel. In contrast, a global kernel allows data points that are far away from each other to have obvious effects on the kernel value as well. A typical example of a global kernel is the polynomial kernel. In this work, we proposed a multikernel function which is taking the advantages of the polynomial kernel function and the Gaussian kernel function. The multikernel function is described as

$$K(x_i,x_j) = (1-\lambda)K_{\mathrm{poly}}(x_i,x_j) + \lambda K_g(x_i,x_j) = \lambda \cdot \exp\left(-g\left\|x_i - x_j\right\|^2\right) + (1-\lambda)\left(x_i \cdot x_j + 1\right)^p. \tag{10}$$

In multikernel function, λ is an adjusting coefficient between K_g and K_{poly}. When λ approximates to 0, the value of K approaches K_{poly}. In the same way, when λ approximates to 1, the value of K approaches K_g. Moreover, the width g and the degree p are the kernel parameters of the Gaussian kernel and the polynomial kernel functions, separately.

For achieving better generalization performance, the four variables λ, g, and p of MKELM and the punish parameter C need to be chosen appropriately.

3.2. QPSO. As an advanced method, QPSO [18] is an intelligence searching method; in the meanwhile, the status of every particle is illustrated by using wave function when all the particles are transferred into the quantum space, rather than the position as well as velocity in traditional PSO [19]. The particles of QPSO could appear much far away from current locations according to characteristics of the wave function, which enhances the possibility of escaping from the local optimal value. Considering that the QPSO algorithms include S-sized particle population in the D-dimensional searching space, the ith particle x_i is renewed based on

$$x_{id} = P_d \pm \delta \cdot \left|m_{\mathrm{best}} - x_{id}\right| \cdot \ln\left(\frac{1}{u}\right), \tag{11}$$

$$P_d = \frac{\varphi_{1d} p_{id} + \varphi_{2d} p_{gd}}{\varphi_{1d} + \varphi_{2d}}, \tag{12}$$

$$m_{\mathrm{best}} = \frac{\sum_{i=1}^{S} P_i}{S} = \left(\sum_{i=1}^{s} \frac{p_{i1}}{s}, \sum_{i=1}^{s} \frac{p_{i2}}{s} \cdots \sum_{i=1}^{s} \frac{p_{iD}}{s}\right). \tag{13}$$

where $d = 1, 2, \ldots, D$, $i = 1, 2, \ldots, S$, μ is a random number distributed equally on (0, 1), δ is a positive number as well as ranges from $\delta_{\min}$ to $\delta_{\max}$ called the contraction expansion (CE) coefficient, which balances the local and the global optimum, P_d is known as the local attractor of every particle at d-dimension on the basis of the trajectory analyses in [22], m_{best} is the mean best position, p_{id} is the best previous position of particle i, p_{gd} is the position of global best particle, and φ_{1d} and φ_{2d} are two different random vectors.

4. PIs Model Construction by QPSO-KELM

According to the theory of PIs, when targets lie in the PI nominal confidence (PINC), the prediction values need to be in the constructed PIs at the possibility of PINC equals $100(1-\alpha)\%$. It can be expressed as

$$t_i \in \left[\widetilde{L}_t^{\alpha}(x_i), \widetilde{U}_t^{\alpha}(x_i)\right], \tag{14}$$

where $\widetilde{L}_t^{\alpha}(x_i)$ and $\widetilde{U}_t^{\alpha}(x_i)$ are the lower borders and upper borders of the prediction PIs at the nominal confidence level of the input x_i.

To construct PIs for the traffic flows, the KELM-based PIs establishing approach can be employed, which is illustrated in Figure 1. The KELM-based probabilistic prediction model targets are shown to directly create the upper borders and lower borders of two outputs. In Figure 1, $U(x)$ and $L(x)$ are the two output vectors of SLFN with respect to the input sample x_i, $\omega_i = [\omega_{i1}, \omega_{i2}, \ldots, \omega_{in}]^T$ is the weight vector connecting the ith hidden node and the input nodes, and $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{in}]^T$ is the weight vector connecting the ith hidden node and the output nodes. Besides, the determination for reliable PIs of the traffic flow via straight optimization for both sharpness and dependability will be demonstrated in the following part.

4.1. Construction of Optimal PIs for Traffic Flow. To assess the properties of the PIs acquired, the PI coverage probability

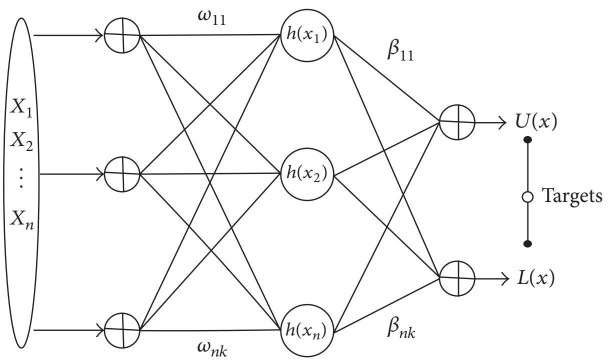


Figure 1: KELM-based PIs construction method.

(PICP) as well as the PI normalized average width (PINAW), which represents the dependability and the sharpness of PIs, is utilized in [11]. PICP is a crucial indicator regarding the dependability for the established PIs, which can be presented by

$$\text{PICP} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} c_i, \tag{15}$$

where N_{test} refers to the number for testing data. If the target t_i locates within the ith lower border and the ith upper border, then $c_i = 1$, or else $c_i = 0$.

In the procedure of interval prediction, the future targets are anticipated to locate within borders of constructed PIs at the PINC level. Nevertheless, it is able to be simply achieved by enlarging PIs from upper or lower bound. Actually, such PIs are meaningless for making a decision. Among the publications, PINAW is defined to indicate the PIs average width quantitatively, which is presented by

$$\text{PINAW} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \left(U_i - L_i\right), \tag{16}$$

where R refers to the targets' range and R is applied to standardize the average width of PIs in terms of percentage.

To guarantee the high properties of created traffic flow's PIs, the output layer parameters β of the MKELM model are improved to lead to higher PICP as well as lower PINAW value. The extensively utilized coverage width-based criterion (CWC) [11] in (11) can be considered for training the KELM model:

$$\text{CWC} = \text{PINAW} \cdot \left(1 + \gamma_{\text{PICP}} \cdot \mathrm{e}^{-\tau(\text{PICP}-\mu)}\right). \tag{17}$$

In (11), μ denotes the confidence probability $100(1-\alpha)\%$. τ works to magnify any tiny discrepancy between PICP and μ, and γ_{PICP} is a function of PICP and defined as

$$\gamma_{\text{PICP}} = \begin{cases} 0, & \text{PICP} > \mu \\ 1, & \text{PICP} < \mu. \end{cases} \tag{18}$$

When PICP is more than the given confidence probability μ, $\gamma_{\text{PICP}} = 0$ and CWC inclines to be identical to PINAW. Otherwise, $\gamma_{\text{PICP}} = 1$ and the function is going to be calculated by CWC. However, it is difficult to decide the value of τ when the CWC function is used.

In this paper, a width deviation rule (WDR) is brought forward to form a new objective function where the exponential punishment term in CWC will be substituted by the equation called WD as follows:

$$\begin{aligned} f(\beta) &= \text{PINAW} + \gamma_{\text{PICP}} \cdot \sigma \cdot \text{WD} \\ &= \text{PINAW} + \gamma_{\text{PICP}} \cdot \sigma \cdot \sum_{i=1}^{N_{\text{test}}} \text{WD}_i, \end{aligned} \tag{19}$$

$$\text{WD}_i = \begin{cases} \dfrac{L(x_i) - t_i}{U(x_i) - L(x_i)}, & \text{if } t_i < L(x_i) \\ 0, & \text{if } t_i \in I(x_i) \\ \dfrac{t_i - U(x_i)}{U(x_i) - L(x_i)}, & \text{if } t_i > U(x_i). \end{cases} \tag{20}$$

The PIs width deviation for each sample can be expressed in (19). σ is set to $1/\alpha^2$ as a penalty coefficient in the present work. For PICP $> \mu$, $f(\beta)$ will be equal to PINAW which is the same as CWC. Otherwise, the width deviation information of all samples is considered as a penalty to describe the sharpness more comprehensively.

4.2. QPSO Algorithm for PIs Optimization. To obtain the effective interval forecasts of traffic flow, the optimal output weights β of MKELM need to be calculated by using QPSO algorithm. The properties of acquired PIs are mirrored by the fitness values of every particle, which is obtained by computing the aim function in (11). The main steps of PIs optimization by using QPSO are demonstrated as follows.

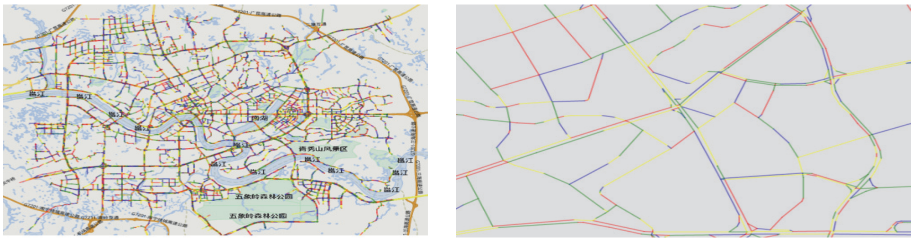
(1) Preprocess. Standardize the training samples and testing samples to $[0, 1]$.

(2) Initialization. The interval prediction model based on MKELM with two outputs needs to be constructed. The upper bounds are set 30% higher than the targets. In the same way, the lower bounds are set 30% lower than the targets. Calculate the initial output matrix of hidden layer β_{int}. The particles position $[x_{i1}, x_{i2}, \ldots, x_{id}]$ is initialized by the initial output weights β_{int}.

(3) Construction of PIs and Evaluation of Cost Function. With the initialization parameters, the PIs model based on MKELM is constructed and the fitness and sharpness are calculated for each particle using (15) and (16).

(4) Renew the Position of Each Particle. All particles' locations are updated in the light of (11)–(13).

(5) Renew P_i and P_g. With the renewed β, establish a novel model and assess the fitness values of fresh particles. If present fitness value is better compared to that of P_i, afterwards P_i will



(a) Overall view of road pieces in Nanning city

(b) Enlarged view of part of road pieces in Nanning city

Figure 2: Roads are split into many small pieces. Road pieces are the basic unit to evaluate or predict.

be renewed. Moreover, if the fitness is better compared to P_g, then P_g will be renewed.

(6) Loop. On condition that the largest amount of repetition has not been achieved, go back to step (4). Or else, the process ends and the optimal output weights are acquired to establish MKELM-based model.

5. Application Studies

5.1. Database. The basic data source comes from traffic information detection system of Traffic Information Center of Nanning, which covers traffic data from main ways, minor roads, and branch roads of Nanning in Figure 2. Data collection is from April 15, 2015, to May 16, 2015. For instance, even if 2,400,000 training data are generated from May 15 to May 16, only 1,515,447 different training data are valid, and the redundancy of it needs to be eliminated. To get an effective database, the original traffic data need to be processed in advance. For example, when the speed is more than 100 km/h, the sample is invalid and needs to be eliminated. When flow, speed, and occupancy are zeros at the same time, the sample also will be removed. Each simulation runs 4-fold cross validation apart from particular account. The training samples are randomly divided into 4 same scale subsamples, where a single one is kept to be the validation data to test the model, and the rest are utilized to be training data every time. The detailed sets on May 15 are illustrated in Table 1.

The 258 attributes include the most important seven kinds of features such as date of week, time of day, traffic density, speed, flow, weather condition, and road logical region in Table 4. Generally, attributes either point out measurements for certain continuous scale such as traffic flow for the last time under this circumstance or demonstrate information about certain definite or separate features such as date type within the above characteristics. The characteristics utilized in this paper blend definite characteristics with real-valued characteristics and are supposed to be converted to all definite or real-valued characteristics. In our conditions, all definite characteristics possess only single value, which is set to be either 0 or 1. For example, characteristic date type from Monday to Friday is going to be converted to 7 new characteristics, each of which refers to either 0 or 1 for pointing out whether it occurs. Therefore, all the characteristics are real-valued and are able to be equally evaluated.

Table 1: Information of data on May 15.

Training data	Testing data	Attributes
1,136,585	378,862	258

5.2. Feature Extraction. The traffic features such as flow parameters, date types, and environmental situations will be described in detail in the following part. Also the most relevant features for high prediction precision are selected by sensitivity analysis.

5.2.1. Feature Description. Actually, the roads are divided into minor road segments as fundamental components for assessment and forecast. The road in Nanning city consists of 18, 041 minor segments. Figure 2(a) illustrates the road segments in Nanning city, and Figure 2(b) refers to the enlarged vision of road segments.

For traffic flow prediction, feature extraction is regarded as a significant part and will impact the forecasting outcomes directly. It is essential to decide appropriate input variables for precise predictions. Apparently, some factors affect the traffic condition of roads, for example, whether there are traffic lights or not, where the road is, and if a school or a shopping center is located nearby. However, it is difficult to obtain these characteristics automatically. Consequently, these characteristics will not be taken into consideration. We chose the following characteristics.

(i) Present time t: categorical, 06:05, 06:10, . . ., 21:55. 191 in total.

Determining how to forecast the traffic flow accurately within a 5 min time interval is critical for ITS because the cycle of transportation induction is usually 5 minutes.

(ii) Date type: categorical, $1, 2, 3, \ldots, 7$, 7 in total.

(iii) Weather condition: categorical, $1, 2, 3, \ldots, 8$, 8 in total; details are shown in Table 2.

(iv) Traffic flow of the last time interval: continuous, normalization data varying from 0 to 1.

$$\text{Volume}^{*} = \frac{\text{Volume} - \text{Volume}_{\min}}{\text{Volume}_{\max} - \text{Volume}_{\min}}, \qquad (21)$$

$$\text{Volume}_{\max} = \frac{1}{n}\sum_{j=1}^{n}\text{Volume}_{\max,j}, \qquad (22)$$

Table 2: Quantificational disposal of weather factors.

Weather condition	Sunny	Cloudy	Windy	Foggy	Rainy	Thunder storm	Snowy	Snow storm
Quantificational value	8	7	6	5	4	3	2	1

$$\text{Volume}_{\min} = \frac{1}{n}\sum_{j=1}^{n}\text{Volume}_{\min,j}, \qquad (23)$$

where $\text{Volume}_{\max,j}$ represents the maximum flow volume of the jth day and $\text{Volume}_{\min,j}$ is the minimum flow volume of the jth day. In addition, n donates the amount of the sample days.

We define $x_{i,j}$ $(0 \leq x_{i,j} < 100)$ as the traffic flow at jth time interval and on the ith road (is collected every 5 minutes and 191 indexes per day totally, from 6:00 to 22:00). For the specific ith road, $[x_{i,0}, x_{i,1}, x_{i,2}, \ldots, x_{i,190}]^T$ is the vector of traffic flow on one day. Thus vectors of all roads are able to be described as follows:

$$\begin{aligned} x_0 &= [x_{0,0}, x_{0,1}, x_{0,2}, \ldots, x_{0,190}]^T \\ x_1 &= [x_{1,0}, x_{1,1}, x_{1,2}, \ldots, x_{1,190}]^T \\ &\vdots \\ x_n &= [x_{n,0}, x_{n,1}, x_{n,2}, \ldots, x_{n,190}]^T . \end{aligned} \qquad (24)$$

(v) Road logical region: categorical, $1, 2, 3, \ldots, 50$, 50 in total.

Physical region, which partly represents the circumstance of roads, can play an important role in road congestion. For example, roads nearby the school could usually stay in a good situation apart from the time when students head for school on mornings and when the students go back home on afternoons. At the same time, traffic situation possesses strong space-time periodicity and area characteristic is supposed to be taken into consideration. Regrettably, the information of physical region cannot be automatically obtained by ITS, and it is easy to transform, particularly in the situation where a new road is in construction when ITS runs. In this paper, a logical region is utilized rather than physical region to demonstrate roads' circumstance characteristics because of the inaccuracy of the area labeled by individuals.

In this paper, the K-means algorithm is used to cluster the logical regions into k clusters. The value of k is set to be 50 in this paper.

(vi) Road type: categorical, 1, 2, 3, 3 in total.

This eigenvalue is defined with the numerical value. From branch way to major road the numerical values are 1, 2, and 3 in order. The higher the road grade, the higher the road's standard speed and maximum speed.

(vii) Number of lanes.

The number of lanes represents the road capacity. Under the circumstances of same traffic flow, the more lanes there are, the higher the speed is.

(viii) The average speed of all cars on the road section.

Firstly, we calculate the average speed of every car in the interval of ΔT on the road section. Then the average speed of all cars is attained. In the interval of ΔT, the velocity measurement sites of the floating car r on the terminal road section are distributed as shown in the figure below.

Table 3: Five levels of the speed of floating cars.

Speed grades	1	2	3	4	5
Range (km/h)	<15	15~35	35~55	55~75	>75

Sequence $\{t_0, t_1, \ldots t_p\}$ and sequence $\{u_0, u_1, \ldots u_p\}$ are floating car's time sequence and speed sequence on the road section. The floating car's driving distance S_r is defined by

$$\begin{aligned} S_r &= \int_{t_0}^{t_p} u\,dt \\ &\approx u_0\left(\frac{t_1 - t_0}{2}\right) + u_p\left(\frac{t_p - t_{p-1}}{2}\right) \\ &\quad + \sum_{i=1}^{p-1} u_i\left(\frac{t_{i+1} - t_{i-1}}{2}\right). \end{aligned} \qquad (25)$$

Therefore, the floating car r's average speed can be represented as $U_r = S_r/(t_p - t_0)$.

If the number of floating cars on the road section at the moment is n, the average speed of all cars on the road section can be written as $\widetilde{U} = (\sum_1^n U_r)/n$.

(ix) The speed distribution of all cars on the road section.

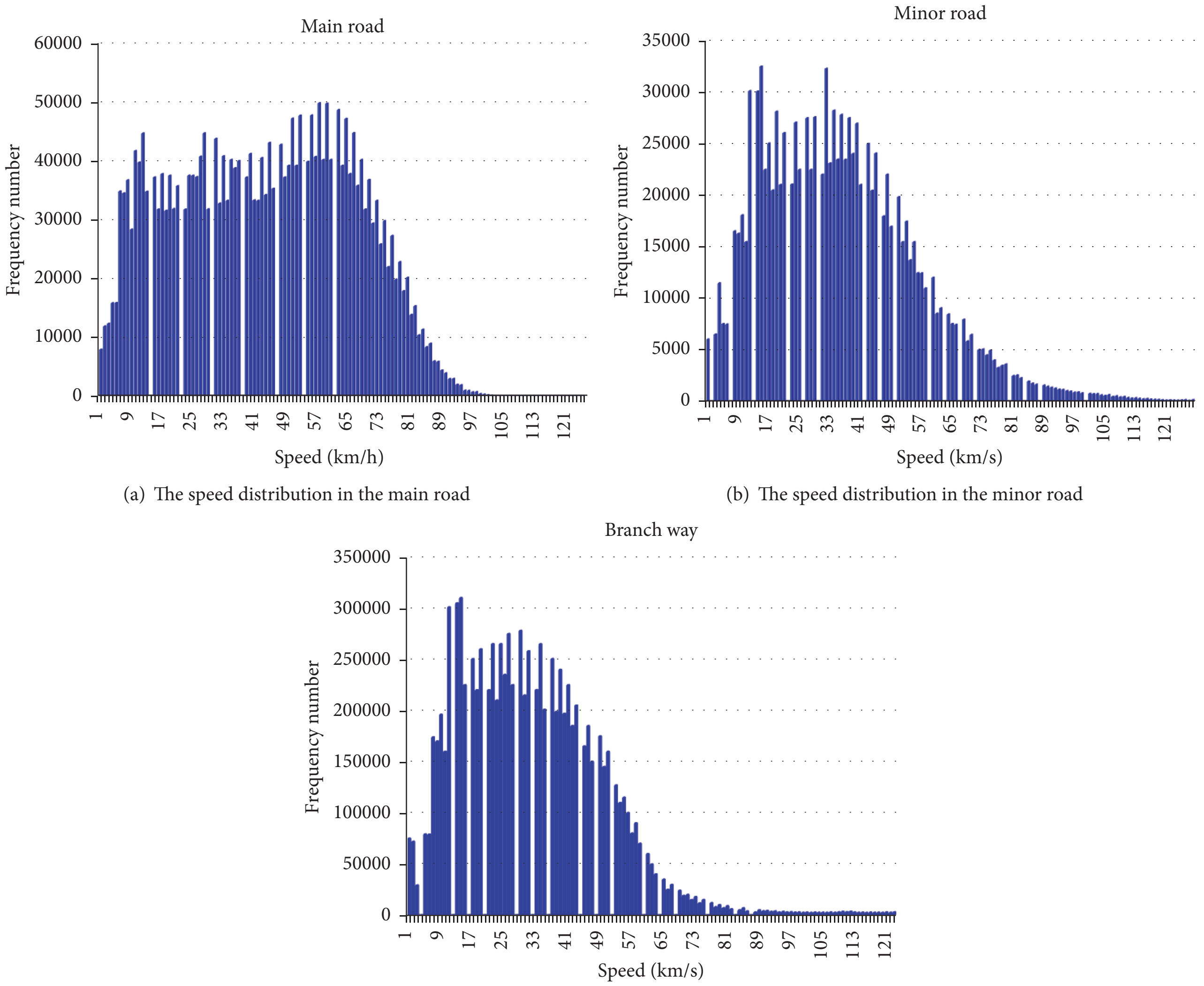
The car speed is distinguished into different levels and the histogram is used to represent the distribution of speed data of all floating cars on the read section. The standard of division is based on the speed distribution of floating cars in the city (Figure 3). Car speed data is mainly no more than 75 km/h. Therefore, the car speed is distributed into 5 grades as shown in Table 3.

(x) The average stopping time of all cars on the road section.

When a car's speed is below 5 km/h, it is identified as a stopped car. The floating car r's stopping time can be represented as $T_r = \sum_{i=0}^{p-1}(t_{I+1} - t_i)(u_i < 5)$.

Thus the average stopping time of all cars on the road section can be written as $\widetilde{T} = (\sum_1^n T_r)/n$.

5.2.2. Input Dimension Reduction. Determining the most relevant features is an issue of research in itself. Taking all the factors into consideration will lead to computation complexity, dimensionality course, and overtraining. There are three sorts of methodologies for decreasing the input



(a) The speed distribution in the main road

(b) The speed distribution in the minor road

(c) The speed distribution in the Branch way

Figure 3: The speed distribution of floating cars in the city on 23rd January 2015.

dimensionality including NNs sensitivity analysis, transformation, and correlation among features [23]. The correlation approaches lead to their high computational complexity. The problem with the transformation approaches is that the features of the original input no longer exist. To address the problems, we selected the sensitivity analysis method as the feature dimension reduction tool in this paper. Sensitivity analysis concentrates on how a network's output is influenced by its input perturbations. Irrelevant inputs are found and eliminated to reduce data collection cost and improve the network's performance. The underlying relationships between the inputs and outputs are found after sensitivity analysis. Sensitivity of an output o_k corresponding to its input x_i is especially

$$S_{ki} = \frac{\partial o_k}{\partial x_i}. \tag{26}$$

Inputs which affect outputs most significantly are determined by examination of the mean square average (MSA) sensitivity matrix $S_{ki,\text{avg}}$. A minor value of $S_{ki,\text{avg}}$ compared with others suggests that, as to the special kth output for the network, the ith input cannot remarkably donate per average to the output k, and consequently could be probably negligible. The one input varies up or below the mean while all other inputs were kept fixed at their respective means. This process was the repeated for each input variable.

The abbreviations of the chosen characteristics are shown in Table 4. And outcomes of sensitivity analysis are presented in Figure 4 in which seven most significant input parameters are date of week (DW), time of day (TD), traffic density (DEN), speed (SP), flow (FL), weather condition (WC), and road logical region (RLR).

When either one or more inputs possess comparably minor sensitivity compared to others, the NNs' dimension could be decreased by eliminating them; meanwhile a smaller-scale NNs are able to be retrained under most circumstances successfully.

Table 4: List of input variables.

Number	Variables	Abbreviation
1	Day of week	DW
2	Time of day	TD
3	Traffic density	DEN
4	Speed	SP
5	Flow	FL
6	Weather condition	WC
7	Road logical region	RLR
8	Road type	RT
9	Number of lanes	NL
10	Speed distribution	SD
11	Stopping time	ST
12	Number of vehicles	NV
13	Number of traffic lights	NTL

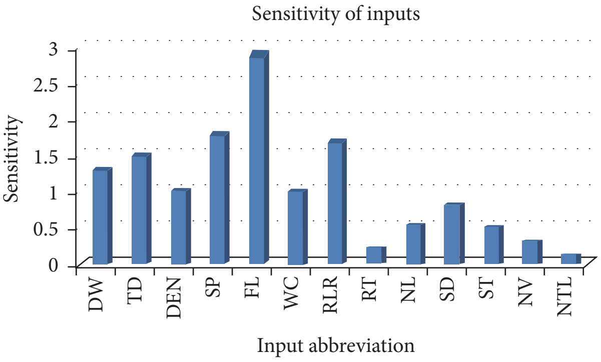


Figure 4: Sensitivity analysis of input variables.

5.3. Determination of Model's Parameters and Its Application to Traffic Congestion Prediction

(a) Optimization of Multikernel Model's Parameters. In (7), it can be observed that the prediction precision of MKELM is mainly affected by the four variables λ, g, and p of MKELM and the punish parameter C. The four variables need to be selected with optimizing method. Then the problem of how to choose proper parameters of MKELM can be converted into optimizing four variables: $x = [g, \lambda, p, C]$. The vector x which has the best fitness is chosen in MKELM. In our study, QPSO is used to obtain optimized parameters. The function for fitness calculation is

$$f(x) = \sqrt{\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \left[t_i - y(x_i)\right]^2}, \tag{27}$$

where $y(x_i)$ is the predicted value, t_i is the target, and m is the amount of training data.

The main procedures are described as follows:

(1) Preprocesses the original samples and divide them into training and testing samples.

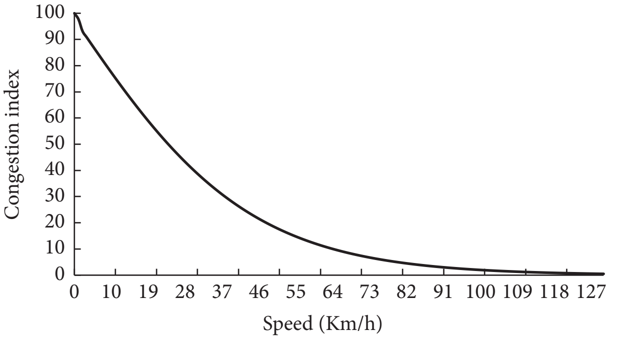


Figure 5: Graph of (21).

(2) Initialize the particles position $[x_{i1}, x_{i2}, \ldots, x_{id}]$. The range of parameters is $x = \{0, 1; 0.04, 100; 0.5, 1.3; 100, 300\}$.

(3) Calculate every particle's fitness of every particle and renew.

(4) Compare and update the top fitness value.

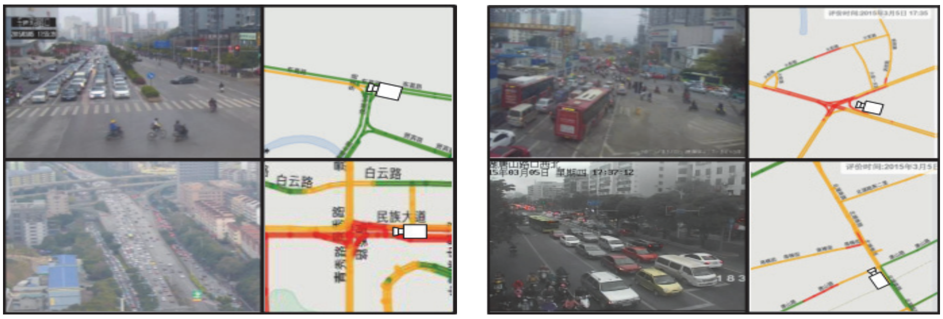
(5) If the ending conditions are met, the optimal parameters of MKELM are obtained.

(b) Traffic Congestion Prediction Using Optimized MKELM Model. Driving speed refers to a significant indicator for judging the road situation. It is simple to obtain average speed of vehicles via gathering 5-minute speed. We primarily utilize speed for determining the Traffic Congestion Indicator (TCI). Equation (28) is the function applied to assess transport situation while its plot is illustrated in Figure 5.

$$y(x) = 100 - 200 \cdot \left(\frac{1}{1 + e^{-0.46x}} - \frac{1}{2}\right), \tag{28}$$

where x refers to the speed of road segment (unit: km/h), whereas $y(x)$ refers to TCI. It can be seen that road situation becomes worst if the average speed turns to be zero, and consequently TCI is equal to 100. Otherwise, traffic situation becomes better as the speed increases. Thus, when speed inclines to be infinite, the value of TCI inclines to be 0, where $y = 0$ is the horizontal asymptote of $y(x)$.

The output of assessment refers to a continuous value which changes between 0 and 100. The feasible result indicates that the speed-TCI model turns out to be rational and be able to represent traffic status appropriately. The assessment outcomes satisfy the true condition as illustrated in Figure 6. In the map, green represents smooth traffic, yellow shows average condition, and red means the road is congested. Seen from the image taken by surveillance cameras, the traffic evaluation accurately reflects the road traffic congestion at that time. The system creates assessment output every 5 minutes so as to portray the present urban traffic situation for every road segment. These data gradually become historical. The historical assessment data are able to be applied to train forecasting model, and it is clear that a large number of training data are generated every day. In practice, almost 1,500,000 valuable data are created every day. Large data call for a proper model with fast speed. It is known that MKELM



(a) (b)

Figure 6: The comparison of camera results and evaluation results shows that the evaluation model is reasonable.

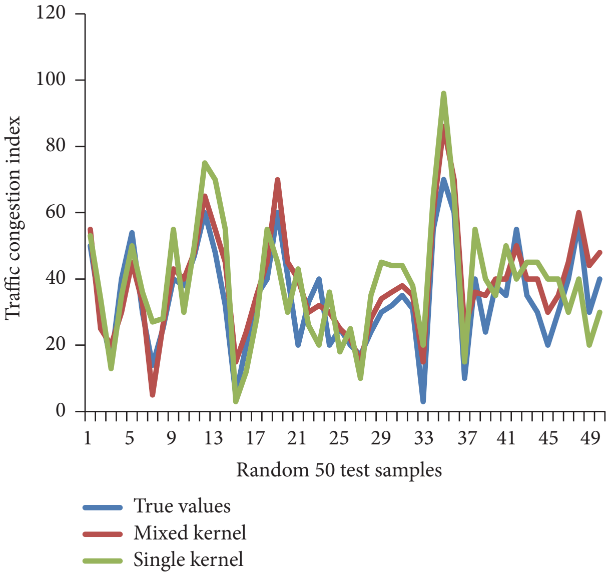


Figure 7: Part of regression prediction shows that mixed kernel model has a great fitting result.

algorithm inclines to offer good generalization feature at a fast learning rate.

With the optimal parameters of MKELM, we just draw target values of 50 testing data selected randomly, single Kernel-ELM predictions, and MKELM, respectively. In standard single kernel model, the radial basis function (RBF) is used of which the parameter γ and the cost coefficient C were selected in $\{20, 21, \ldots, 210\}$. Figure 7 shows that the multikernel model of KELM has more accurate prediction results on traffic training data than standard one-kernel ELM. The difference between the true values and the ones predicted by multikernel model are very small for most of the cases.

5.4. Study Results and Discussions. In this experiment, the kernel function of KELM is multikernel described in (10). For the parameters of QPSO, the population scale as well as the amount of repetitions is 200; the values of $\delta_{\min}$ and $\delta_{\max}$ are set to be 0.5 and 0.9.

The main target of the proposed QPSO-KELM approach is to originate dependable PIs. Moreover, traffic management needs valuable information along with higher confidence levels. Consequently, it is supposed to be more virtually significant to acquire high-confidence-level PIs for satisfying the requirements of traffic management. Diverse levels of PINC $100(1-\alpha)\%$ from 80% to 95% are taken into account in this research.

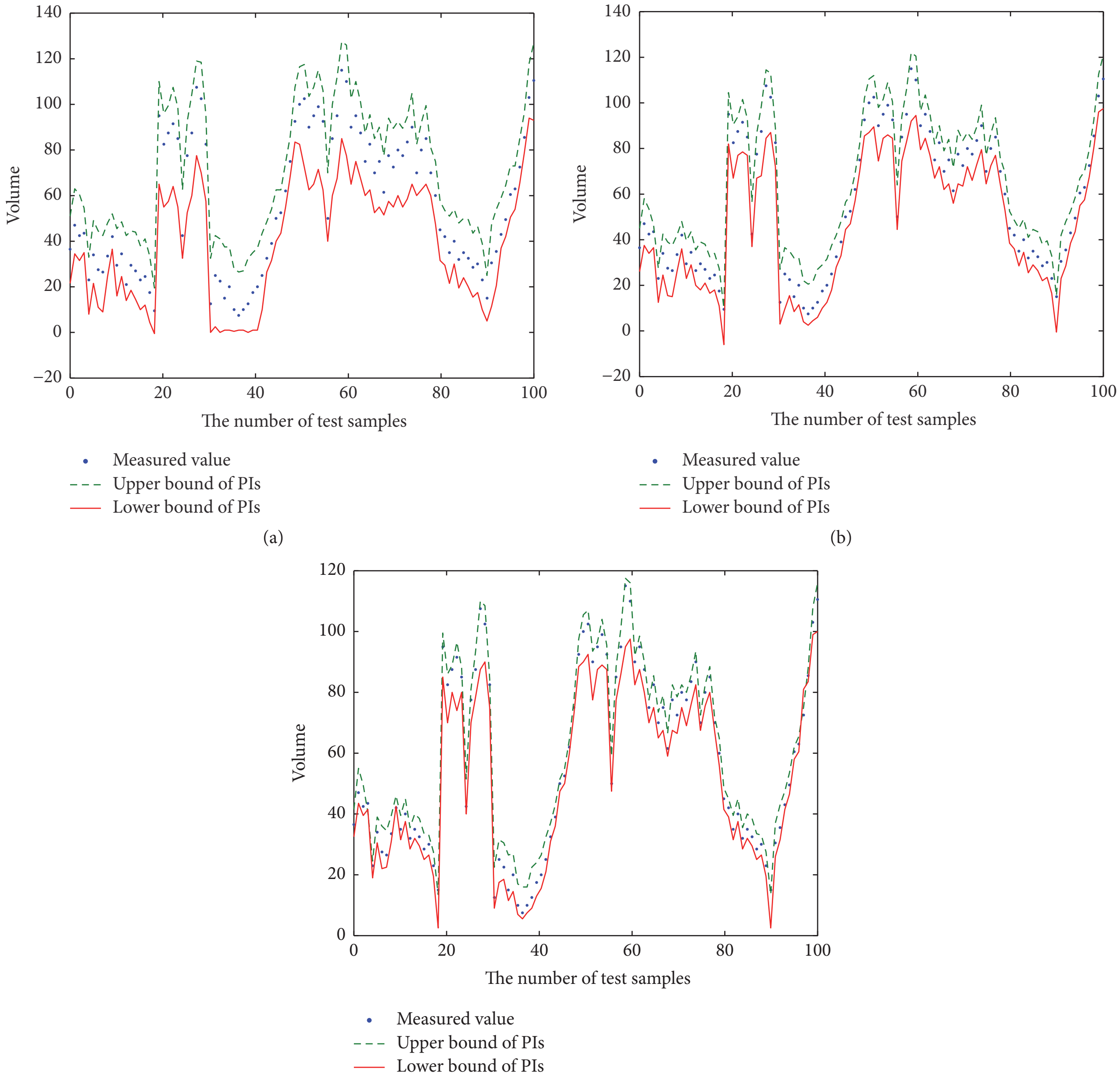
Table 5: Comparison of MKELM-CWC and MKELM-WDR.

PINC%	Methods	PICP (%)	PINAW (%)
95%	MKELM-CWC	95.86	28.97
	MKELM-WDR	95.37	25.09
90%	MKELM-CWC	91.04	22.61
	MKELM-WDR	90.49	21.45
80%	MKELM-CWC	80.93	14.78
	MKELM-WDR	80.46	13.97

(1) Comparison of MKELM-Based Model Using CWC and WDR. To construct the PIs for traffic flow, whether the rule CWC or the rule WDR should be used in the MKELM-QPSO model needs to be decided firstly.

For the traffic flow, the dependability indicator PICP as well as the sharpness indicator PINAW of the constructed PIs at three different confidence levels 80%, 90%, and 95% is summarized in Table 5. Also the results using CWC and WDR are, respectively, summarized in Table 5. It is shown that, at every confidence level, the PICP values of WDR are closer to the confidence levels than the values of CWC. At the same time, CWC has a much higher sharpness than WDR which provides much more accurate PIs. The comparisons of CWC and WDR in Table 5 show that WDR generates the higher dependability and sharpness of the PIs. In the rest of the paper, the criterion WDR is applied to get the optimal PIs because of its better performance shown in Table 5.

To achieve a better visual effect, we draw the real values and forecasting PIs, respectively, of 100 testing samples which are selected randomly. The constructed optimal PIs at different confidence levels 80%, 90%, and 95% are shown in Figure 8. Most of the targets have been limited within the upper and the lower bound. For different confidence levels, most of the measured traffic flows are limited in the PIs established by the proposed method, which means that the



(a)

(b)

(c)

Figure 8: Overall optimal PIs by proposed QPSO-KELM approaches. (a) 95% confidence level case. (b) 90% confidence level case. (c) 80% confidence level case.

remarkable performance will satisfy the requirement of traffic management.

Also the nonstationary and nonlinearity characteristics of traffic flow are clearly demonstrated in these graphs.

(2) Comparison of Different Methods for Constructing the PIs. For the sake of further demonstrating the effectiveness of the QPSO-MKELM method, different intelligence search and learning algorithms are compared using the same dataset.

As to the PIs dependability test, corresponding PICPs and PINAWs are demonstrated in Table 6. The corresponding PICPs and PINAWs outcomes of diverse methods are described in Table 6. At different confidence levels, the dependability indicator PICP and the sharpness indicator PINAW are summarized in Table 6, which are acquired by QPSO-MKELM, PSO-MKELM, and GA-MKELM. From the outcomes of PICP as well as PINAW indices, it is shown that QPSO-KELM methods offers far more precise PIs combined dependability (indicated by higher values of PICP) as well as sharpness (reflected by lower values of PINAW) on the three various confidence levels.

The SVM, ELM, standard KELM, and MKELM are compared on the basis of the training time, PICP, and PINAW at the same confidence level 90% in Table 7. Although the

Table 6: Comparison of QPSO-KELM, PSO-KELM, and GA-KELM.

PINC%	QPSO-MKELM		PSO-MKELM		GA-MKELM	
	PICP%	PINAW%	PICP%	PINAW%	PICP%	PINAW%
95	95.37	25.09	94.32	30.54	93.86	31.07
90	90.49	21.45	89.54	29.17	88.93	29.66
80	80.46	13.97	79.37	22.89	78.86	23.85

Table 7: Comparison of SVM, ELM, KELM, and MKELM at confidence level 90%.

Algorithms	Training times (s)	PICP%	PINAW%
SVM	19204.86	90.56	23.43
ELM	300.39	87.34	37.57
KELM	811.05	90.84	21.76
MKELM	860.83	90.49	21.45

MKELM cannot compete with the ELM in terms of the training time, the MKELM provides much more accurate PIs than the ELM. We can also see that the SVM almost provides the same PICP and PINAW as the MKELM, but the training time for the SVM is more than 20000% of that for the KELM. Overall, the MKELM outperforms the SVM, the ELM, and the standard KELM. Thus, it is rational to draw a conclusion that the proposed method is an efficient probabilistic forecasting approach for traffic flow.

6. Conclusions

In order to develop an effective probabilistic forecasting method for traffic flow, a novel method on the basis of QPSO-MKELM has been described to establish the reliable PIs. MKELM has been developed to establish the reliable PIs, and the parameters of MKELM are optimized by using QPSO. The seven features including date of week, time of day, traffic density, speed, flow, weather condition, and road logical region are selected as inputs of KELM by sensitivity analysis. The experimental results have shown that QPSO-MKELM is an effective method to establish the optimal PIs. Moreover, the proposed method can offer far more precise PIs that combined higher dependability and sharpness at different confidence levels than other methods. Additionally, successful utilization in practical traffic flow prediction shows that QPSO-MKELM is an effective probabilistic forecasting method. In the current paper, only limited traffic flows are taken into consideration; other parameters such as accidents, traffic jams, or seasonal variation have not been discussed. In future study, more possible conditions will be included in a longer period of time.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (no. 61272357, no. 61300074, and no. 61572075) and the National Key Research and Development Program of China (Grant no. 2016YFB0700502 and Grant no. 2016YFB1001404). The authors thank Dr. Ruoyi Liu for revising their English expressions.

References

[1] D. Ngoduy and A. Sumalee, "Adaptive estimation of noise covariance matrices in unscented Kalman filter for multiclass traffic flow model," *Transportation Research Record*, vol. 2188, pp. 119–130, 2010.

[2] R. K. Oswald, W. T. Scherer, and L. B. Smith, "Forecasting using approximate nearest neighbor nonparametric regression," Traffic Flow, 2000.

[3] D. Park, L. R. Rilett, and G. Han, "Forecasting multiple-period freeway link travel times using neural networks with expanded input nodes," in *Proceedings of the International Conference on Applications of Advanced Technologies in Transportation Engineering*, 2010.

[4] B. L. Smith and M. J. Demetsky, "Short-term traffic flow prediction: neural network approach," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1453, pp. 98–104, 1994.

[5] L. I. Bin, X. I. Tao, and M.-H. Shi, "Traffic flow combined forecast model of Support Vector Machine," *Journal of Tianjin Polytechnic University*, vol. 27, no. 2, pp. 73–76, 2008.

[6] N. L. Nihan and K. O. Holmesland, "Use of the box and Jenkins time series technique in traffic forecasting," *Transportation*, vol. 9, no. 2, pp. 125–143, 1980.

[7] Y.-N. Yang and H.-P. Lu, "Short-term traffic flow combined forecasting model based on SVM," in *Proceedings of the International Conference on Computational and Information Sciences (ICCIS '10)*, pp. 262–265, IEEE, Chengdu, China, December 2010.

[8] P. Pinson and G. Kariniotakis, "Conditional prediction intervals of wind power generation," *IEEE Transactions on Power Systems*, vol. 25, no. 4, pp. 1845–1856, 2010.

[9] C. A. M. da Silva Neves, M. Roisenberg, and G. S. Neto, "A method to estimate prediction intervals for artificial neural networks that is sensitive to the noise distribution in the outputs," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '09)*, pp. 2238–2242, IEEE, Atlanta, Ga, USA, June 2009.

[10] A. Khosravi, E. Mazloumi, S. Nahavandi, D. Creighton, and J. W. C. Van Lint, "Prediction intervals to account for uncertainties

in travel time prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 537–547, 2011.

[11] D. A. Nix and A. S. Weigend, "Estimating the mean and variance of the target probability distribution," in *Proceedings of the IEEE World Congress on Computational Intelligence International Conference on Neural Networks*, vol. 1, pp. 55–60, 1994.

[12] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Probabilistic forecasting of wind power generation using extreme learning machine," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1033–1044, 2014.

[13] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower upper bound estimation method for construction of neural network-based prediction intervals," *IEEE Transactions on Neural Networks*, vol. 22, no. 3, pp. 337–346, 2011.

[14] G.-B. Huang and C.-K. Slew, "Extreme learning machine: RBF network case," in *Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision (ICARCV '04)*, pp. 1029–1036, Kunming, China, December 2004.

[15] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.

[16] J. Kivinen, A. J. Smola, and R. C. Williamson, *Learning with Kernels*, MIT Press, 2002.

[17] Y. Xing, X. Ban, C. Guo, and Y. Wang, "Probabilistic forecasting of traffic flow using kernel based extreme learning maching and quantum-bahaved particle swarm optimization," in *Proceedings of the 4th International Conference on Cloud Computing and Intelligence Systems (CCIS '16)*, pp. 205–209, Beijing, China, August 2016.

[18] H. Q. Wang, F. C. Sun, Y. N. Cai, N. Chen, and L. G. Ding, "On multiple kernel learning methods," *Acta Automatica Sinica*, vol. 36, no. 8, pp. 1037–1050, 2010.

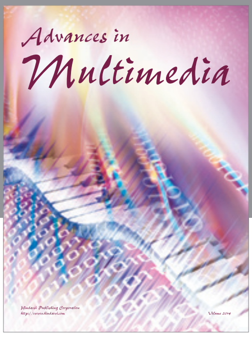
[19] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, pp. 1571–1580, Portland, Ore, USA, June 2004.

[20] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

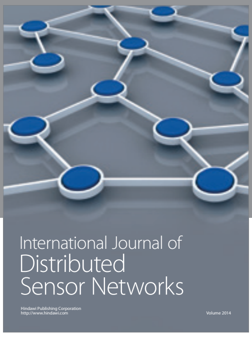
[21] J. Mercer, "Functions of positive and negative type, and their connection with the theory of integral equations," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 209, pp. 415–446, 1909.

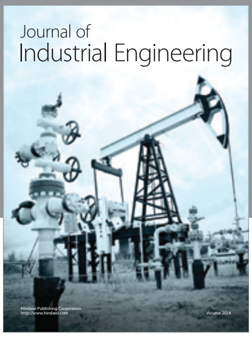
[22] F. van den Bergh and A. P. Engelbrecht, "A study of particle swarm optimization particle trajectories," *Information Sciences*, vol. 176, no. 8, pp. 937–971, 2006.

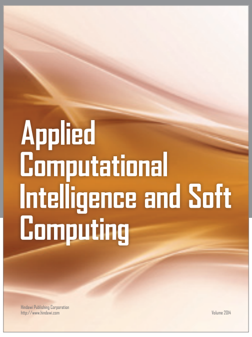
[23] D. S. Yeung, I. Cloete, D. Shi, and W. Ng, *Sensitivity Analysis for Neural Networks*, Natural Computing Series, Springer, Berlin, Germany, 2009.

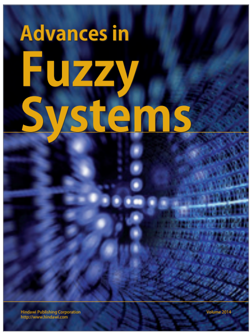

Advances in
Multimedia

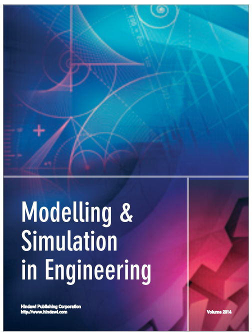
The Scientific
World Journal

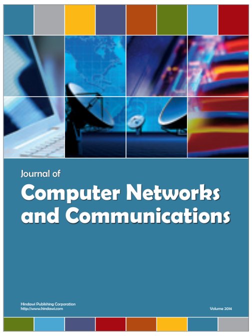

International Journal of
Distributed
Sensor Networks

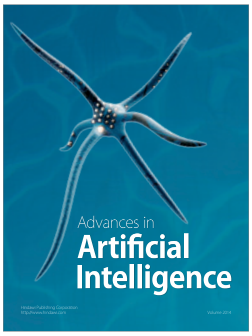

Journal of
Industrial Engineering

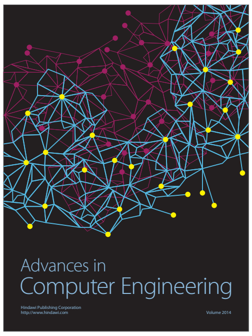

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

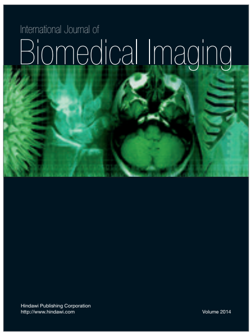

Modelling &
Simulation
in Engineering

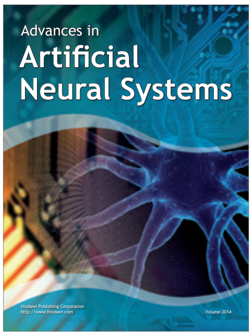

Journal of
Computer Networks
and Communications

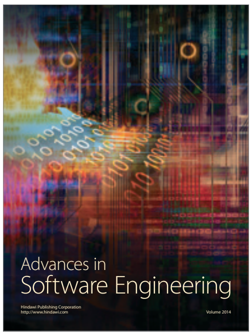

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

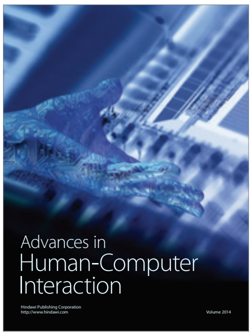
International Journal of
Computer Games
Technology

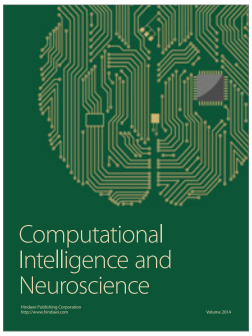

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering