RESEARCH ARTICLE

A Hybrid Short-Term Traffic Flow Prediction Model Based on Singular Spectrum Analysis and Kernel Extreme Learning Machine

Qiang Shang[1], Ciyun Lin[1,2]*, Zhaosheng Yang[1,2], Qichun Bing[1], Xiyang Zhou[1]

1 College of Transportation, Jilin University, Changchun 130022, China, **2** State Key Laboratory of Automobile Simulation and Control, Jilin University, Changchun 130022, China

* linciyun@jlu.edu.cn

Abstract

Short-term traffic flow prediction is one of the most important issues in the field of intelligent transport system (ITS). Because of the uncertainty and nonlinearity, short-term traffic flow prediction is a challenging task. In order to improve the accuracy of short-time traffic flow prediction, a hybrid model (SSA-KELM) is proposed based on singular spectrum analysis (SSA) and kernel extreme learning machine (KELM). SSA is used to filter out the noise of traffic flow time series. Then, the filtered traffic flow data is used to train KELM model, the optimal input form of the proposed model is determined by phase space reconstruction, and parameters of the model are optimized by gravitational search algorithm (GSA). Finally, case validation is carried out using the measured data of an expressway in Xiamen, China. And the SSA-KELM model is compared with several well-known prediction models, including support vector machine, extreme learning machine, and single KLEM model. The experimental results demonstrate that performance of the proposed model is superior to that of the comparison models. Apart from accuracy improvement, the proposed model is more robust.

Citation: Shang Q, Lin C, Yang Z, Bing Q, Zhou X (2016) A Hybrid Short-Term Traffic Flow Prediction Model Based on Singular Spectrum Analysis and Kernel Extreme Learning Machine. PLoS ONE 11(8): e0161259. doi:10.1371/journal.pone.0161259

Editor: Zhaohong Deng, Jiangnan University, CHINA

Received: May 1, 2016

Accepted: August 2, 2016

Published: August 23, 2016

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This work was supported by the Chinese National High Technology Research and Development Program (Grant No. 2014BAG03B03), the National Science Foundation of China (Grant No. 51408257 and No. 51308248). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

1 Introduction

Short-term traffic flow prediction is one of the most important issues in the field of intelligent transport system (ITS) [1–2]. Many subsystems of ITS such as Advanced Traffic Management System (ATMS) and Advanced Traveler Information Systems (ATIS) can benefit from improved prediction of traffic flow parameters (such as traffic volume, average traffic speed, and average occupancy) in a short-term future.

Many researchers have paid more attention to short-term traffic flow prediction because of its importance. As a result, a large number of relevant methods have been published in the academic literature. In general, these methods are categorized into three types [3–4]: traffic model-based methods, statistical methods and machine learning-based methods. A brief summary of these three types of methods is shown in Table 1.

Table 1. A brief summary of the three types of methods.

Types	Advantages	Disadvantages
Traffic model-based methods (macroscopic traffic model, microscopic traffic model)	Well-established theoretical background and good interpretability, consider the effect of various factors (incidents, road works, traffic control measures, etc)	Many parameters and assumptions need to defined in advance, parameter calibration is difficult, small errors of these parameters could lead to large prediction errors
Statistical methods (local linear regression, ARIMA)	comparatively basic structure, easy and practical to be used	Requires a great deal of historical data, gives some restrictive assumptions, difficult to model nonlinear traffic flow
Machine learning-based methods (ANN, SVM)	Gains knowledge from training data, do not need to predefined model structure, good robustness, ability to approximate any degree of complexity of traffic flow	Requires a training procedure and a large number of training data, poor interpretability

doi:10.1371/journal.pone.0161259.t001

1. Traffic model-based methods, which use the traffic flow model (include macroscopic model and microscopic model) to predict the evolving mechanism of traffic flow. Macroscopic model regards the traffic flow as fluid for exploring its evolution mechanism based on fluid dynamics [5–6]. Microscopic model focuses on behavior of single vehicle and the interactions between vehicles, such as lane changing model [7] and car following model [8].
2. Statistical methods, which is implemented by using the historical data to obtain optimal parameters in the fitting process. Typical statistical methods have been proposed and applied for years, such as local linear regression model [9] and Auto-Regressive Integrated Moving Average (ARIMA) [10].
3. Machine learning-based methods, which can effectively capture the nonlinearity relationship between the input and output existed in the data by using intelligent learning algorithm. Such as Artificial Neural Network (ANN) [11–12] and Support Vector Machine (SVM) [13–14].

Reader interested in details of these prediction models can refer to review papers such as Refs. [15–17]. By analyzing the characteristics of the three types of methods, traffic model-based methods and statistical methods often assume several restrictive assumptions, such as a predefined model structure, the normality of residuals and the stationary of the time series, which are seldom satisfied in the case of nonlinear and chaotic traffic flow. However, machine learning-based methods are not restricted by previous assumptions. Particularly, ANN and SVM are more popular because of their mature theoretical basis and the excellent prediction performance.

In recent years, a novel learning algorithm called Extreme Learning Machine (ELM) is proposed for training Single Layer Feed-forward Neural Network (SLFN) [18]. In ELM, input weights and hidden biases are assigned randomly instead of being exhaustively tuned. For this reason, ELM training is fast and saves a lot of computing resources. Moreover, ELM also avoids falling into local minima in the learning process. Therefore, ELM achieves superior performance than traditional SLFN and Back-Propagation (BP) learning algorithm. Because of these advantages of ELM, it has been applied in different fields successfully [19–25]. In order to improve the generalization ability of ELM and reduce time consumption for determining the number of hidden layer nodes, Kernel Extreme Learning Machine (KELM) is developed more recently by replacing the ELM hidden layer with a kernel function [26]. Different with SVM, the kernel function of KELM does not need to satisfy Mercer's theorem. Thus application of KELM is easier than SVM. In addition, KELM is applied to solve many prediction or classification problems and achieves comparative or superior performance [27–30]. In spite of great success in some scientific fields, the KELM method is not utilized for predicting short-term traffic flow at present. Thus, we have reasons to believe that KELM method has a better application prospect in traffic flow prediction and try to employ KELM method for building a traffic flow prediction model in this paper. At the same time, to obtain the optimal KELM model, it is important to choose a kernel function and determine the model parameters. If the parameters are not selected correctly, the performance of KELM model is greatly reduced. In this paper, Gravitational Search Algorithm (GSA) is introduced to optimize the parameters of KELM model, which is a heuristic optimization method based on the Newtonian gravity law [31]. The GSA has demonstrated potential in parameter determination for nonlinear models [32–34].

Moreover, the input form of KELM model affects its performance and training speed. With the development of chaos theory, recent studies such as Refs. [35–37] have concluded that the nonlinear chaotic phenomena existed in short-term traffic flow time series. Phase Space Reconstruction (PSR) is the basis of chaotic time series analysis, which affects the prediction performance directly. It can map scalar time series to the multi-dimensional phase space and mining the implicit information thoroughly. Therefore, PSR is used to determine the optimal input form of KELM model in this paper. Delay time and embedding dimension are the key parameters for PSR. At present, there are many methods for choosing the two parameters. The methods can be used to calculate delay time including Autocorrelation Function method [38], Mutual Information method [39] and Average Displacement method [40]. The methods can be used to calculate embedding dimension including G-P method [41], False Nearest Neighbors method [42] and Cao method [43]. However, C-C method [44] is different from these methods and can simultaneously estimate delay time and embedding dimension based on statistical results. The advantages of C-C method are small amount of calculation, strong anti-interference ability and easy to use. Thus, we choose C-C method to determine the two parameters for PSR.

The data pre-processing is necessary before training the KELM model. Due to inevitable interference in the process of data acquisition or transmission, the original traffic data contain some noise components and unpredictable components (outliers) which have less useful information and often lead to over-fitting. In pre-processing process, removing noise and outliers of chaotic time series can reduce its complexity and increase its predictability [45]. Spectrum Analysis (SSA) is a relatively new time series analysis technique which combines multivariate statistic, probability theory and signal processing [46]. SSA is suitable for time series with various features and structures, such as stationary and non-stationary, linear and nonlinear time series [47]. It can extract the main features of time series, remove noise and unpredictable components effectively. So far, SSA has received increasing attentions and has been employed in several areas successfully, such as medicine, energy, climatology, and economics [48–51]. Therefore, we choose SSA to filter traffic flow time series in the pre-processing process.

The main objective of this work is to propose a hybrid short-term traffic flow prediction model named SSA-KELM using SSA and KELM. The KELM can achieve a better prediction performance using the filtered traffic flow time series data, because noise components of the original traffic time series have been removed by SSA during the pre-processing process. The novelty of the proposed model is highlighted in the following aspects.

1. The KELM as a relatively novel machine-learning method is first studied for short-term traffic flow prediction.
2. The SSA as an effective data pre-processing method is employed to process the original traffic flow series.
3. In view of the chaotic characteristics of traffic flow, PSR is performed to determine the optimal input form of the KELM.

4. The parameters of KELM are tuned by gravitational search algorithm (GSA) to achieve a better prediction performance.
5. Non-parametric statistical tests are used for multiple comparisons of different prediction models on multiple data sets.

The rest of this paper is organized as follows. In Section 2, SSA method and KELM model are described briefly, and the hybrid traffic flow prediction model (SSA-KELM) is introduced. In Section 3, empirical analysis is performed, the prediction results of several different prediction models are given and discussed. In Section 4, conclusions and recommendations for future study are presented.

2. Methodology

In this setction, the basic methods related to the SSA-KELM model are briefly introduced, including Singular Spectrum Analysis (SSA), Kernel Extreme Learning Machine (KELM), Phase Space Reconstruction (PSR) and Gravitational Search Algorithm (GSA). Then the main steps of SSA-KELM model are given. In the SSA-KELM model, KELM is the basic prediction method, SSA is used to filter the noise components from the original traffic data, PSR is used to determine the optimal input form of KELM and the GSA is used to optimized the paramenters of the KELM.

2.1 Singular spectrum analysis (SSA)

SSA performs four steps including embedding, singular value decomposition, grouping and diagonal averaging. The first two steps are usually named decomposition of time series, while the third step and fourth step called reconstruction of time series. A brief review of SSA is as follows (more information can be obtained in Ref. [52]).

Step 1: Embedding. This step is to transform the original series into a sequence of multi-dimensional vector. For one dimensional series vector with length N as $\mathbf{X} = (x_1, \cdots, x_N)$. Given a window length $L(1 < L < N)$, the initial series is mapped into K lagged vectors:

$$\mathbf{X}_i = (x_i, x_{i+1}, \cdots, x_{i+L-1})^T, i = 1, 2, \cdots, K = N - L + 1 \tag{1}$$

Then, the trajectory matrix is expressed as Eq (2). The matrix $\mathbf{T}$ is a Hankel matrix with size of $L \times K$, which has equal elements x_{ij} along the anti-diagonals where $i + j$ = const,

$$\mathbf{T} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K] = \begin{bmatrix} x_1 & x_2 & \cdots & x_K \\ x_2 & x_3 & \cdots & x_{K+1} \\ \cdots & \cdots & \cdots & \cdots \\ x_L & x_{L+1} & \cdots & x_N \end{bmatrix} \tag{2}$$

Step 2: Singular Value Decomposition (SVD). Let $\mathbf{S} = \mathbf{TT}^T$, the eigenvalues of $\mathbf{S}$ are calculated and arranged in decreasing order, denoted as $\lambda_1 \geq \cdots \geq \lambda_L \geq 0$. The eigenvectors of matrix $\mathbf{S}$ corresponding to these eigenvalues are denoted as $\mathbf{U}_1, \cdots \mathbf{U}_L$. Then the SVD of trajectory matrix $\mathbf{X}$ is defined as Eq (3):

$$\mathbf{T} = \mathbf{T}_1 + \cdots + \mathbf{T}_d \tag{3}$$

where d is the rank of $\mathbf{T}$, $\mathbf{T}_i = \sqrt{\lambda_i}\mathbf{U}_i\mathbf{V}_i^T (i = 1, \cdots, d)$ are elementary matrix (rank is 1), $\mathbf{V}_i = \mathbf{T}^T\mathbf{U}_i/\sqrt{\lambda_i}$ are principal components of matrix $\mathbf{T}$. The collection $(\sqrt{\lambda_i}, \mathbf{U}_i, \mathbf{V}_i)$ is termed as ith eigen triple of SVD. The contribution of $\mathbf{T}_i$ can be measured by ratio of eigenvalues and given by Eq (4),

$$\eta_i = \lambda_i / \sum_{i=1}^{d} \lambda_i \tag{4}$$

Step 3: Grouping. Indices set $\{1, \cdots, d\}$ is divided into m disjointed subsets $I_1, \cdots, I_m$. Let $I = \{i_1, \cdots, i_p\}$ and the resultant trajectory matrix $\mathbf{T}_I$ corresponding to the group I is defined as $\mathbf{T}_I = \mathbf{T}_{i_1} + \cdots + \mathbf{T}_{i_p}$. The resultant trajectory matrices are calculated for every group $I = I_1, \cdots, I_m$ and the expansion of Eq (3) leads to the decomposition as follows.

$$\mathbf{T} = \mathbf{T}_{I_1} + \cdots + \mathbf{T}_{I_m} \tag{5}$$

Step 4: Diagonal averaging: the grouped decomposition in Eq (5) is transformed a new series with length N. Let $\mathbf{Y}_1 = (y_1, \cdots, y_N)$ be the transformed one dimensional series of $\mathbf{T}_{I_1}$, elements of $\mathbf{Y}_1$ is calculated by Eq (6).

$$y_k = \begin{cases} \frac{1}{k}\sum_{j=1}^{k} y^*_{j,k-j+1}, 1 \le k \le L^* \\ \frac{1}{L^*}\sum_{j=1}^{L^*} y^*_{j,k-j+1}, L^* \le k \le K^* \\ \frac{1}{N-k+1}\sum_{j=k-K+1}^{N-K^*+1} y^*_{j,k-j+1}, K^* \le k \le N \end{cases} \tag{6}$$

where $L^* = \min(L, K)$, $K^* = \max(L, K)$; if $L < K$, $y^*_{j,k-j+1} = y_{j,k-j+1}$ and if $L \ge K$, $y^*_{j,k-j+1} = y_{k-j+1,j}$. Thus the original series $\mathbf{X} = (x_1, \cdots, x_N)$ is decomposed into m series:

$$\mathbf{Y} = \mathbf{Y}_1 + \cdots + \mathbf{Y}_m \tag{7}$$

2.2 Kernel extreme learning machine (KELM)

A brief description of ELM and KELM is given in this section. Interested readers can refer to Refs. [18, 26] for more details. The output function of ELM for generalized SLFN is

$$f_L(x) = \sum_{i=1}^{L} \beta_i h_i(x_j) = \sum_{i=1}^{L} \beta_i g_i(w_i, b_i, x_j) = h(x)\boldsymbol{\beta}, j = 1, \cdots, N \tag{8}$$

where $g_i(\cdot)$ denotes the output of the i-th hidden node with respect to the input x, i.e., activation function, such as "Sigmoid" function. $w_i \in R^n$ is the weight vector connecting the ith hidden layer neuron and the input layer neurons, b_i the bias of the ith hidden layer neuron, $\beta_i \in R$ is the weight connecting the ith hidden neuron and the output neuron, and $f_L(x)$ is the output of the SLFN. w_i and b_i are randomly assigned before learning.

If SLFN can approximate theses N samples with zero error, there exists β_i, w_i and b_i such that

$$\sum_{i=1}^{L} \beta_i g(w_i, b_i, x_j) = t_j, j = 1, 2, \ldots, N \tag{9}$$

Thus Eq (9) is expressed compactly as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \tag{10}$$

Where

$$\mathbf{H} = \begin{pmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{pmatrix} = \begin{bmatrix} h(w_1, b_1, x_1) & \cdots & h(w_L, b_L, x_1) \\ \vdots & \ddots & \vdots \\ h(w_1, b_1, x_N) & \cdots & h(w_L, b_L, x_N) \end{bmatrix}_{N \times L} \tag{11}$$

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{pmatrix}_{L \times m} \text{ and } \mathbf{T} = \begin{pmatrix} t_1^T \\ \vdots \\ t_N^T \end{pmatrix}_{N \times m} \tag{12}$$

Where **H** is the hidden layer output matrix, **β** is output weight matrix, and **T** is the target matrix. In the ELM, **β** is the only parameter that needs to be calculated and can be easily achieved by Least Squares Estimate (LSE)

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \tag{13}$$

Where $\mathbf{H}^{\dagger}$ is Moore-Penrose generalized inverse of matrix **H**. one of the methods for calculating $\mathbf{H}^{\dagger}$ is the orthogonal projection method:

$$\mathbf{H}^{\dagger} = \mathbf{H}^T \left(\mathbf{H}\mathbf{H}^T\right)^{-1} \tag{14}$$

According to the ridge regression theory, the diagonal of $\mathbf{H}\mathbf{H}^T$ can be add a positive value for regularization, then we have

$$f(x) = h(x)\boldsymbol{\beta} = h(x)\mathbf{H}^T \left(\frac{\mathbf{I}}{C}\mathbf{H}\mathbf{H}^T\right)^{-1} \mathbf{T} \tag{15}$$

Where **I** is a unit matrix, C is penalty parameter. When the hidden feature mapping function $h(x)$ is unknown, a kernel matrix for ELM is used according to the following equation:

$$\Omega_{\text{ELM}} = \mathbf{H}\mathbf{H}^T : \Omega_{\text{ELM}_{i,j}} = h(x_i) \cdot h\left(x_j\right) = K\left(x_i, x_j\right) \tag{16}$$

where $K(x_i, x_j)$ is a kernel function. Thus, the output function of ELM is denoted compactly as

$$f(x) = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}^T \left(\frac{\mathbf{I}}{C} + \Omega_{\text{ELM}}\right)^{-1} \mathbf{T} \tag{17}$$

Because of the application of kernel function in ELM, this novel learning algorithm is named KELM. In KELM, weight vector w_i, bias b_i, the feature mapping function $h(x)$ and the number of hidden layer neurons L are not taken into consideration. Instead, KELM only focuses on the kernel functions $K(x_i, x_j)$ and the training data set. Many kernel functions can be used for KELM, such as linear, polynomial and Gaussian radial basis function (RBF). We choose the Gaussian RBF kernel $K(u, v) = \exp(-\|u - v\|^2 / 2\sigma^2)$ to construct KELM model because of its superior performance [26, 30]. In the KELM with Gaussian RBF kernel, the are two key parameters (penalty parameter C and kernel parameter σ) need to be determined.

2.3 Phase space reconstruction (PSR)

According to Takens' Embedding Theorem [53], as enough delayed coordinates are used, scalar time series is sufficient to reconstruct the dynamic of the underlying systems. For a time series $\{x(i), i = 1, 2, \cdots, N\}$, the phase space can be reconstructed according to

$$X(i) = \{x(i), x(i+\tau), \cdots, x[i+(m-1)\tau]\} \tag{18}$$

Where τ is delay time, m is embedding dimension. According to phase space reconstruction. Input of the model is $[x_i, x_{i+\tau}, \cdots, x_{i+(m-1)\tau}]$ and output of the model is $x_{i+(m-1)\tau+1}$.

The C-C method is used to determine delay time τ the embedding dimension m. According to Ref. [39], the basic principle of C-C method is as follows.

The correlation integral is defined as

$$C(m, N, r, t) = \frac{2}{M(M-1)} \sum_{1 \leq i \leq j \leq M} H\left(r - d_{ij}\right) \tag{19}$$

where r is the search radius, $M = N–(m– 1)\tau$ is the number of embedded points in m-dimensional space, N is the data number of the time series. $d_{ij} = \|X_i–X_j\|$ is Euclidean distance between two points, $H(z)$ is a Heaviside function

$$H(z) = \begin{cases} 1 & (z > 0) \\ 0 & (z \leq 0) \end{cases} \tag{20}$$

The correlation integral is a cumulative distribution function and denotes the probability of distance that between any pairs of points in the phase space is not greater than r.

The time series $\{x(i), i = 1, 2, \cdots, N\}$ is divided into t disjoint time series:

$$\begin{array}{c} \{x(1), x(t+1), x(2t+1), \cdots\} \\ \{x(2), x(t+2), x(2t+2), \cdots\} \\ \vdots \\ \{x(3), x(t+3), x(2t+3), \cdots\} \end{array} \tag{21}$$

The test statistics is

$$S(m, N, r, t) = \sum_{l=1}^{t} \frac{1}{t} \left[C_l(M, \frac{N}{r}, r, t) - C_l^m\left(1, \frac{N}{r}, r, t\right) \right] \tag{22}$$

As $N \rightarrow \infty$, the Eq (22) can be expressed as

$$S(m, r, t) = \sum_{l=1}^{t} \frac{1}{t} \left[C_l(M, r, t) - C_l^m(1, r, t) \right] \tag{23}$$

The maximum deviation $\Delta S(m, t)$ of $S(m, r, t) \sim t$ with r is defined as

$$\Delta S(m, t) = \{\max S(m, r_i, t) - \min S\left(m, r_j, t\right)\} \tag{24}$$

According to the BDS statistical results obtained by Brock et al. [54], $m = 2, 3, 4, 5$, $r_i = i\sigma / 2$ and $i = 1, 2, 3, 4$ is selected in general, where σ is standard deviation of time seires. Calculate the following variables,

$$\bar{S}(t) = \frac{1}{16}\sum_{m=2}^{5}\sum_{j=1}^{4} S(m, r_j, t) \tag{25}$$

$$\Delta\bar{S}(t) = \frac{1}{4}\sum_{m=2}^{5} \Delta S(m, t) \tag{26}$$

$$Scor(t) = \Delta\bar{S}(t) + |\bar{S}(t)| \tag{27}$$

where $\bar{S}(t)$ is the mean of $S(m, r, t)$ for all subsequence. Let t equal or be smaller than 200, the first local minimum point of $\Delta\bar{S}(t) \sim t$ is the delay time τ. The global minimum point of $Scor(t) \sim t$ is the delay time window τ_w. The embedding dimension m is calculated according to $\tau_w = (m - 1)\tau$.

2.4 Gravitational search algorithm (GSA)

In GSA, a set of agents has been given to find optimum solution based on Newtonian gravity law. Assume agents with number N in the dimension of n, and the position of ith agent is defined as:

$$X_i = \left(x_i^1, \cdots, x_i^d, \cdots, x_i^n\right), i = 1, 2, \cdots, N \tag{28}$$

Where x_i^d is the position of ith agent in the dth dimension. The main steps of GSA are as follows:

Step 1: Initialize velocity and positon of each agent.

Step 2: Evaluate the fitness of each agent.

Step 3: Update gravitational constant $G(t)$ according to the following equation:

$$G(t) = G_0 * e^{-\alpha/T} \tag{29}$$

where G_0 is the initial gravitational constant, α is decay rate (some arbitrary constant) and T is the maximum number of iterations.

Step 4: Calculate the gravitational (inertia) mass of each agent using the following equation:

$$\begin{cases} m_i(t) = \dfrac{fit_i(t) - worst(t)}{best(t) - worst(t)} \\ M_i(t) = \dfrac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \end{cases} \tag{30}$$

Where $M_i(t)$ and $fit_i(t)$ are the mass and the fitness value of the agent i at time t, respectively. $worst(t)$ and $best(t)$ are the worst fitness value of agent and the best fitness value of agent, respectively.

Step 5: Calculate the resultant force of agents according to the law of universal gravitation.

$$F_{ij}^d(t) = G(t)\frac{M_i(t) \cdot M_j(t)}{R_{ij}(t) + \varepsilon} \cdot \left(x_i^d(t) - x_j^d(t)\right) \tag{31}$$

where $F_{ij}^d(t)$ is the gravitation between the agent i in the d dimension and agent j at time t; $M_i(t)$ and $M_j(t)$ are the passive gravitational mass of agent i and agent j respectively; ε is the small constant for avoiding the divisor equal to zero; $x_i^d(t)$ is the position of agent *i* in *d* dimension and $x_j^d(t)$ is the position of agent *j* at time *t*; $R_{ij}(t) = \|x_i(t), x_j(t)\|_2$ is the Euclidean distance between agent *i* and agent *j*.

Step 6: Calculate accelerated velocity of agent. Accelerated velocity $a_i^d(t)$ at time *t* in *d* dimension is expressed as follows.

$$a_i^d(t) = \frac{\sum_{i=1,j\neq 1}^{N} rand \cdot F_{ij}^d(t)}{M_i(t)} \tag{32}$$

where *rand* is a random number in the interval [0, 1].

Step 7: Update the positon and velocity of each agent according to Eqs (33) and (34) respectively.

$$v_i^d(t+1) = rand \cdot v_i^d(t) + a_i^d(t) \tag{33}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{34}$$

Step 8: If the stopping criterion is satisfied, output optimal parameters, else proceed to Step 2.

2.5 SSA-KELM model

The overall flowchart of SSA-KELM model is illustrated as Fig 1. And the main steps of SSA-KELM model are as follows.

Step 1: The original traffic flow time series is filtered by SSA and get a reconstruction time series without nosie components.

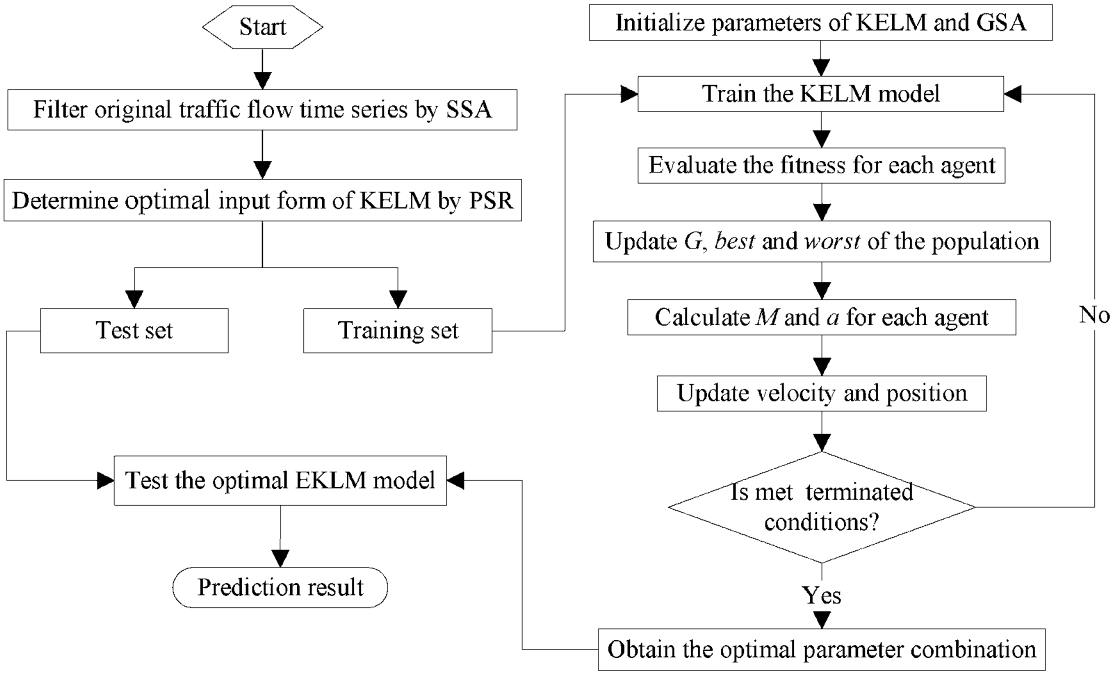
Start
Filter original traffic flow time series by SSA
Determine optimal input form of KELM by PSR
Test set
Training set
Test the optimal EKLM model
Prediction result
Initialize parameters of KELM and GSA
Train the KELM model
Evaluate the fitness for each agent
Update G, best and worst of the population
Calculate M and a for each agent
Update velocity and position
Is met terminated conditions?
No
Yes
Obtain the optimal parameter combination

Fig 1. The flowchart of SSA-KELM Model.

doi:10.1371/journal.pone.0161259.g001

Step 2: The optimal input form of the KELM is determined by PSR. The filtered traffic flow time series is used to train the KELM model, while the original traffic flow time series is used to test the KELM model.

Step 3: The parameters of KELM model are optimized by GSA.

(3.1) Initialize the parameters of the KELM model and GSA.

(3.2) Train the KELM model and evaluate the fitness value of each agent.

(3.3) Update $G(t)$, $best(t)$ and $worst(t)$ of the population.

(3.4) Calculate $M_i(t)$ and $a_i^d(t)$ for each agent.

(3.5) Update the positon and velocity of each agent.

(3.6) Judge whether the terminated conditions are reached (usually the default calculation accuracy or iterations). If reached, continue to step 3.7; otherwise, go to step 3.2 and continue to search.

(3.7) Output the optimal parameters of KELM model.

3 Empirical research

In this section, the measured data is used to validate the performance of SSA-KELM model. First, a detailed description of the data source is given. Then, the traffic data is used to illustrate how to construct and optimize the SSA-KELM model. Finally, the performance of the SSA-KELM model is compared to that of several prediction models on multiple data sets. Besides the traditional statistical indices are employed to evaluate the performances of all the prediction models, non-parametric tests are used to analyze the results of the contrast experiment.

3.1 Data Source

All the experimental traffic volume data is obtained from two loop detectors (No. DC00004965 and No. DC00004966) installed on an expressway named Lianqian W Rd in Xiamen, China. The two detectors' locations are approximately shown in Fig 2, where No. DC00004965 detector locates in the westbound direction while No. DC00004966 detector locates in the eastbound direction. The traffic data is collected every 5 min in five consecutive working days (January 5, 2015 to January 9, 2015). Although average speed is also available, only traffic volume is considered in this study. The original traffic volume data is shown in Fig 3. The traffic volume data of the first four days is used to construct the SSA-KELM model, while the traffic volume data of the fifth day is used to test the KELM model. In the following several subsections, traffic volume data from No. DC00004966 detector is used as an example to illustrate how to build and optimize the SSA-KELM model in detail, while traffic volume data from the two detectors is all used to evaluate the performance of the SSA-KELM model.

3.2 SSA decomposition and reconstruction

As mentioned in subsection 2.1, the window length L is the only parameter in the decomposition process. If the time series has a periodic component, the window length is taken proportional to that period to get better separability [55]. Therefore, $L = 288$ is assumed here, which corresponds to daily variations of traffic volume time series. This window length results in 288 Eigen triples. Each Eigen triples corresponds to a singular value. Fig 4 depicts the curve of leading 72 singular values in descending order (take the base-10 logarithm). As shown in Fig 4, the leading 31 singular values decrease rapidly while the remaining singular values decrease slowly,

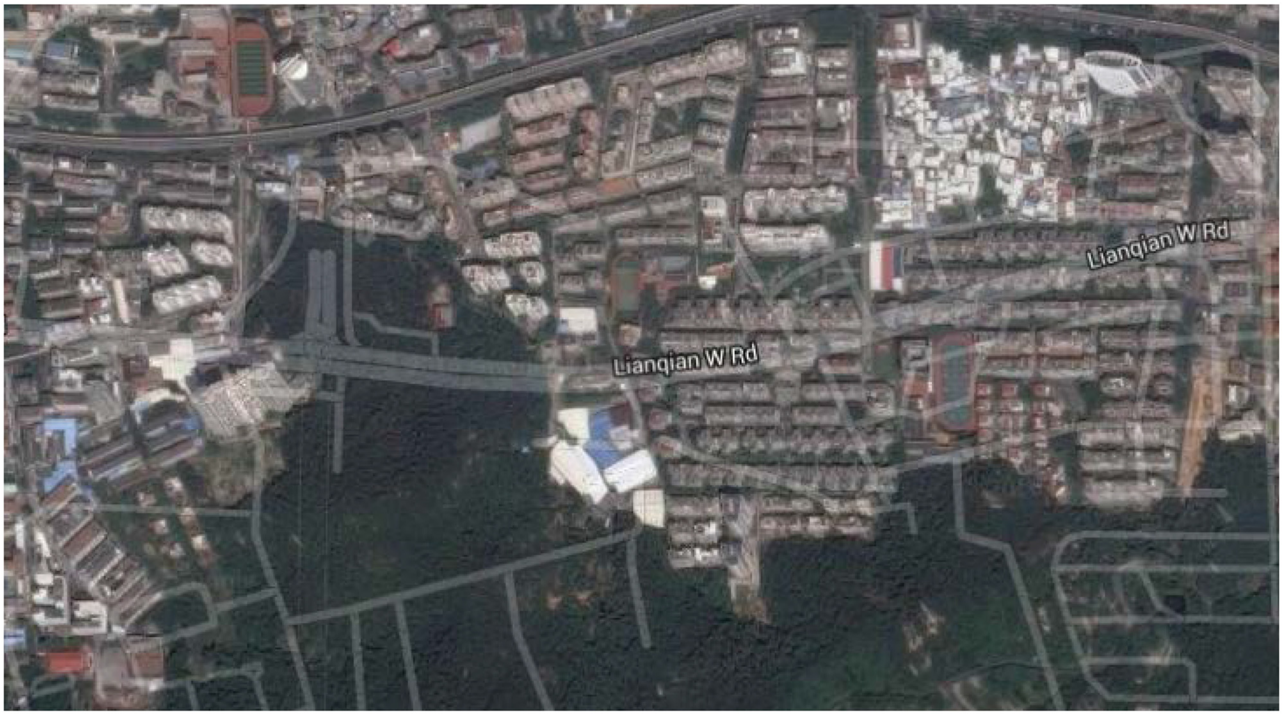

Fig 2. The two detectors' locations on the Lianqian W Rd.

doi:10.1371/journal.pone.0161259.g002

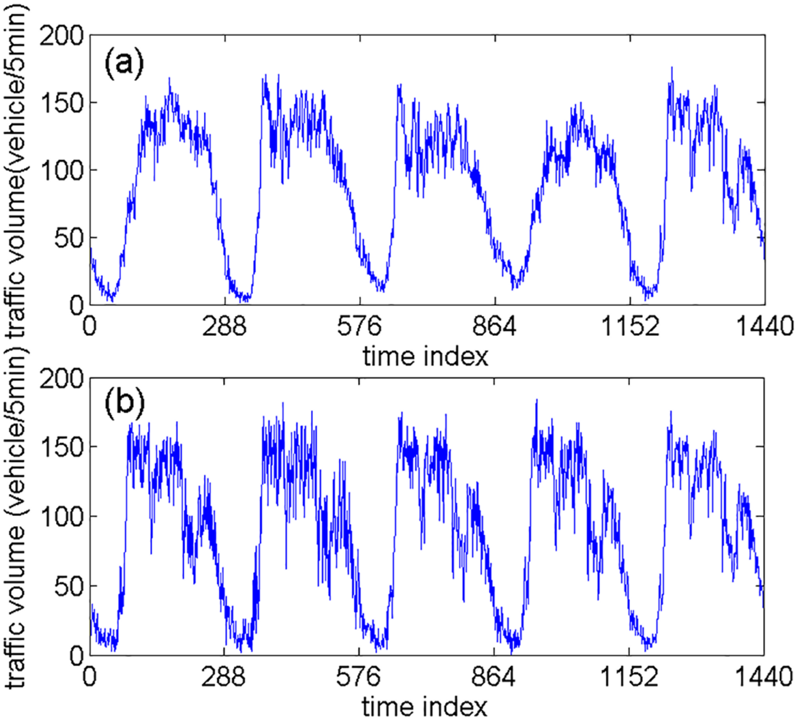

Fig 3. The traffic volume data in five consecutive working days: (a) DC00004965 and (b) DC00004966.

doi:10.1371/journal.pone.0161259.g003

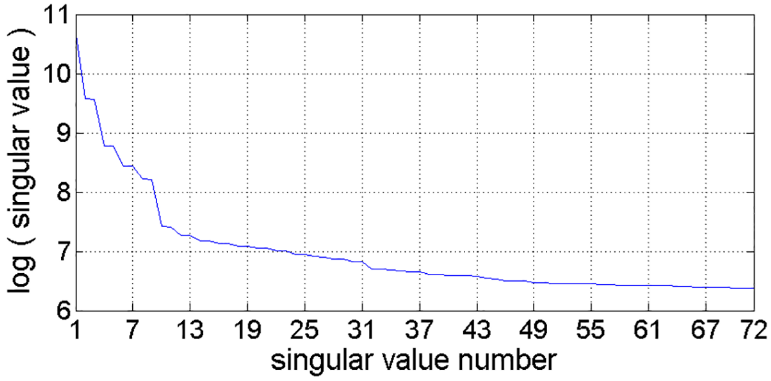

Fig 4. Singular values curve (in descending order).

doi:10.1371/journal.pone.0161259.g004

which means that the remaining eigen triples correspond to the noise components and can be ignored in the reconstruction process. Moreover, the singular value pairs with same or similar singular value, such as {2, 3}, {4, 5}, {6, 7} and . . ., which correspond to oscillatory components.

Fig 5 shows the reconstruction traffic volume time series and contribution ratio corresponding to the leading 9 Eigen triples. The first Eigen triple almost represents the whole trend of the original traffic volume series, which has a share (contribution) of 75.812% of original traffic volume series. Every oscillatory component produces two Eigen triples with close singular values, such as {2, 3}, {4, 5}, {6, 7}, {8, 9}, which have a close frequency and a close contribution ratio.

According to the above analysis, the traffic volume series is reconstructed by using the leading 31 Eigen triples. Fig 6 shows the reconstruction series with a share of 98.27% of original traffic volume series. Fig 7 shows the noise (residual) series with a share of 1.73% of original traffic volume series. As shown in Fig 7, we can see it clearly that the reconstruction series has a satisfactory residual.

3.3 Determination of the optimal input form

The C-C method is used to implement phase space reconstruction for determining the optimal input form of the proposed model. Fig 8 shows the $\Delta\bar{S}(t) \sim t$ curve. The variable t

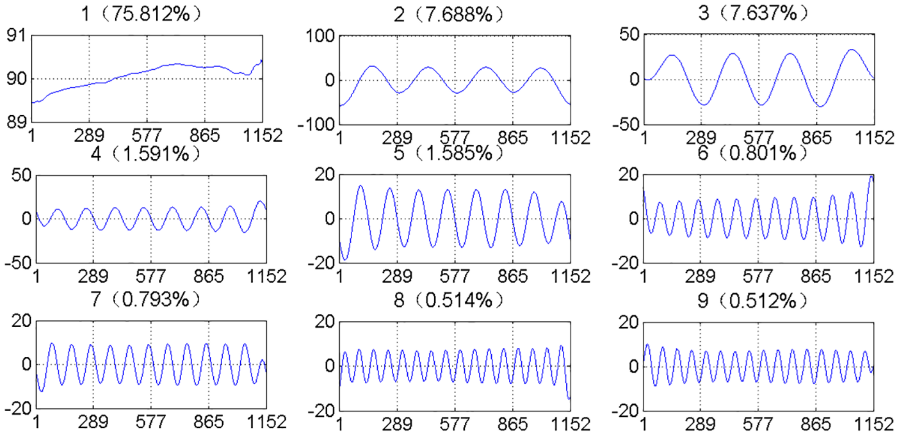

Fig 5. The reconstruction series and contribution ratio corresponding to the leading 9 Eigen triples.

doi:10.1371/journal.pone.0161259.g005

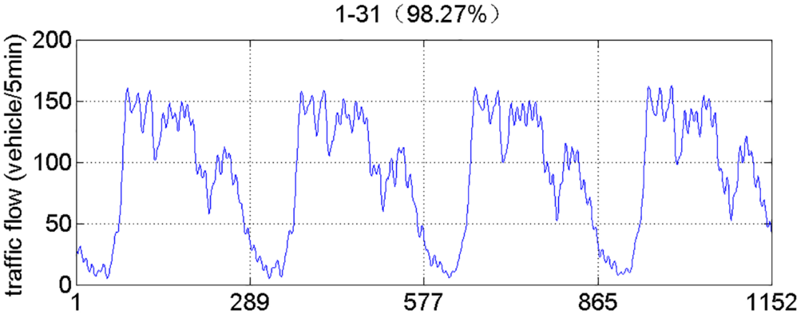

Fig 6. The reconstruction series.

doi:10.1371/journal.pone.0161259.g006

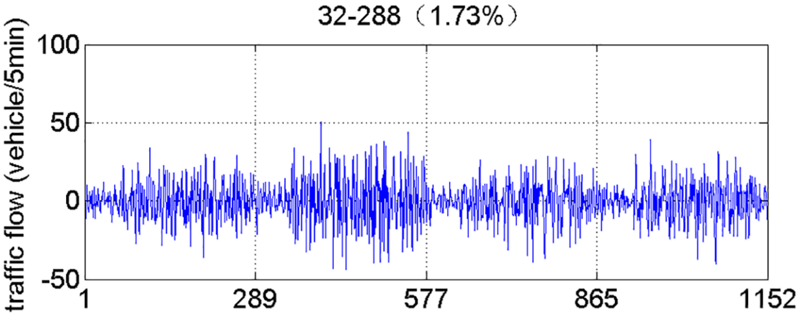

Fig 7. The noise (residual) series.

doi:10.1371/journal.pone.0161259.g007

corresponding to the first local minimum value of $\Delta\bar{S}(t)$ is the time delay τ and we can get $\tau = 9$. Fig 9 shows $Scor(t) \sim t$ curve. The variable t corresponding to the global minimum value of $Scor(t)$ is the embedding window width τ_w of the time series and we can get $\tau_w = 125$. Thus, the embedding dimension $m = 15$ according to $\tau_w = (m-1)\tau$. Based on phase space reconstruction, input and output of the training set are as follows.

$$\text{input} = \begin{bmatrix} x_1 & x_{1+9} & x_{1+2\times9} & \cdots & x_{1+(15-1)\times9} \\ x_2 & x_{2+9} & x_{2+2\times9} & \cdots & x_{2+(15-1)\times9} \\ x_3 & x_{3+9} & x_{3+2\times9} & \cdots & x_{3+(15-1)\times9} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{1025} & x_{1025+9} & x_{1025+2\times9} & \cdots & x_{1025+(15-1)\times9} \end{bmatrix}$$

$$\text{output} = \begin{bmatrix} x_{1+(15-1)\times9+1} \\ x_{2+(15-1)\times9+1} \\ x_{3+(15-1)\times9+1} \\ \vdots \\ x_{1025+(15-1)\times9+1} \end{bmatrix}$$

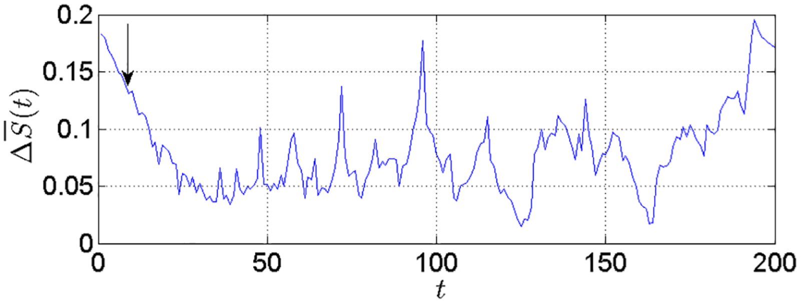

Fig 8. The $\Delta\bar{S}(t) \sim t$ curve.

doi:10.1371/journal.pone.0161259.g008

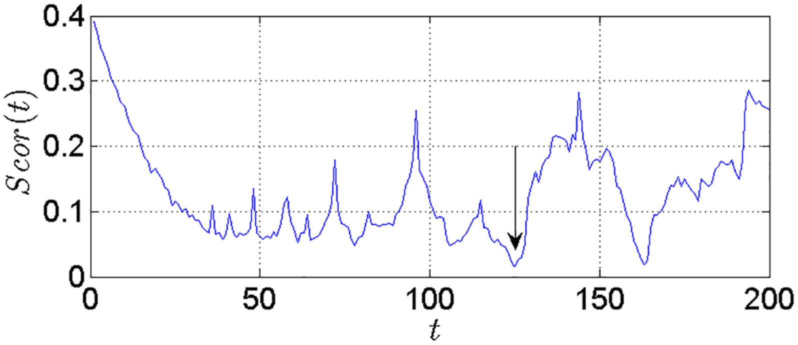

Fig 9. The *Scor*(*t*) ~ *t* curve.

doi:10.1371/journal.pone.0161259.g009

3.4 Parameter Optimization

GSA is employed to optimize the two parameters of the KELM model. In order to ensure the algorithm efficiency, the mean absolute percentage error is taken as fitness function. Then the fitness function is calculated as follows.

$$fitness = \frac{1}{n}\sum_{i=1}^{n} \left|\frac{y_i - \tilde{y}_i}{y_i}\right| \tag{35}$$

Where y_i is the real value in time interval i, $\tilde{y}_i$ is the prediction value for time interval i and n is the total number in the time series.

The specific parameters of GSA are as follows: population size is set to 20, the initial gravitational constant $G_0 = 100$, the attenuation rate $\alpha = 10$, the maximum number of iteration is set to 100, the value range of parameter C is set to [0.1, 1000] and the value range of kernel parameter σ is set to [0.01, 100].

In the process of parameter optimization, the k-cross validation method [56] is used to train the model. Because this method can make full use of the information in the sample and avoid over-fitting and under-fitting. In other words, it can improve the generalization ability of the model under the premise of ensuring good prediction accuracy. In k-cross validation, the training data set is randomly divided into k subsets. The $k - 1$ subsets are used as training set for building the model and the kth subset is used as a validation set for verifying model performance. Each subset is used as a validation set and the verifying is repeated k times in total. The average value of the results of k times verifying is used to evaluate the model performance. Fig 10 shows the fitness curves (include the best fitness and the average fitness). As shown in Fig 10,

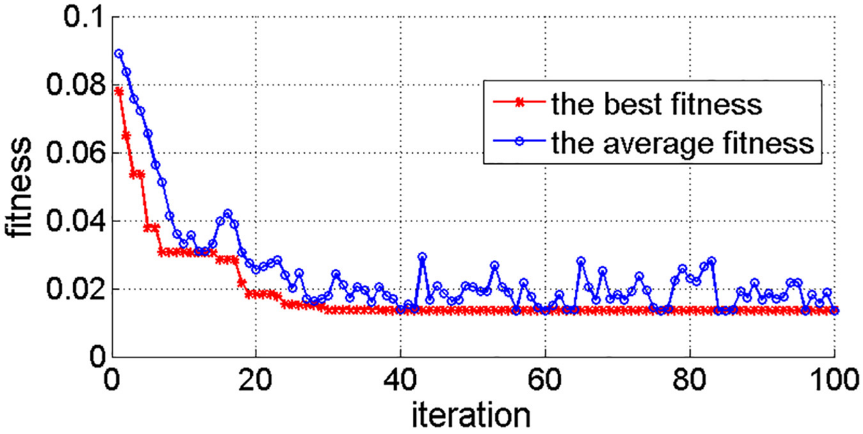

Fig 10. Fitness curves of GSA.

doi:10.1371/journal.pone.0161259.g010

with the increase of the number of iterations, the fitness curve is gradually convergent. When the number of iterations is 96, both the best fitness and average fitness are 0.0135 (MAPE = 1.35%). And the corresponding optimal parameter combination of the model is $C = 7.28$ and $\sigma = 0.15$.

3.5 Performance Evaluation Index

In order to evaluate the performance of the proposed model, two statistical indices are utilized to measure the prediction accuracy. These indices are the MAE (mean absolute error) and MAPE (mean absolute percent error). The smaller values of MAE and MAPE, the more accurate prediction results are. The calculation formulas for the MAE and MAPE are as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \tilde{y}_i| \tag{36}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{y_i - \tilde{y}_i}{y_i}| \tag{37}$$

where y_i is the real value in time interval i, $\tilde{y}_i$ is the prediction value for time interval i and n is the total number of the time series.

3.6 Model Performance and Analysis

Traffic volume data from the fifth day is used as test set to evaluate the performance of the proposed model. For illustrating the model performance intuitively, Figs 11 and 12 show prediction results of the proposed model using NO. DC00004965 detector's data and NO. DC00004966 detector's data, respectively. Figs 11(a) and 12(a) present the curves of prediction data and measured data. The prediction curves can fit the measured curves well. Figs 11(b) and 12(b) present the scatterplots between the predicted data and the measured data. It is clear that these scatter points distribute near the measured line (the red line) without large deviation. Figs 11(c) and 12(c) show APE (absolute percent error) of the predicted data. The APE are mostly within 15%. However, the APE from 24 to 48 (correspond to 2:00–4:00) is larger, and the reason is that the actual traffic volume data is too small during that period.

To illustrate the superiority of the SSA-KELM model, contrast experiment is carried out. In this paper, the related models for comparsion are as follows: the single model (KELM), the

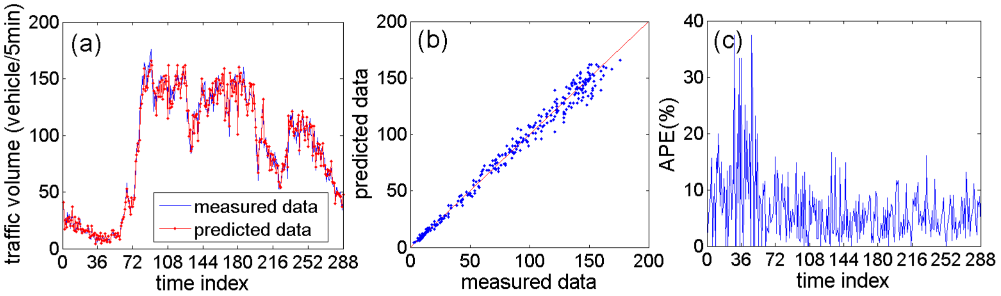

Fig 11. The prediction performance of the proposed method based on NO. DC00004965 detector's data: (a) Curves of predicted data and measured data; (b) Scatterplot of predicted data and measured data; (c) Absolute percent error of predicted data.

doi:10.1371/journal.pone.0161259.g011

hybrid model (SSA-SVM) based on SSA and SVM, and the hybrid model (SSA-ELM) based on SSA and ELM. The optimal input forms of all models are determined by PSR, and the training sets of SSA-SVM model and SSA-ELM model are the same as the SSA-KELM model. However, the original traffic volume data without being filtered by SSA is used to train the KELM model. The parameters of the related models are optimized by GSA, shown in Table 2.

Besides related models (SSA-SVM, KELM and SSA-ELM) mentioned above, two state-of-the-art traffic flow prediction models are used for comparison, that are Hybrid Particle Swarm Optimization Support Vector Regression (HPSO-SVR) model [57] and Long Short-Term Memory Neural Network (LSTM-NN) model [58]. In the contrast experiment, we implement the two state-of-the-art models according to detail steps of the corresponding references.

Fig 13(a) describes the prediction results of the related models based on NO. DC00004965 detector's data, Fig 13(b), 13(c) and 13(d) are specific parts of the prediction results. Fig 14(a) describes the prediction results of the state-of-the-art models based on NO. DC00004965 detector's data. Fig 14(b), 14(c) and 14(d) are specific parts of the prediction results. Fig 15(a) describes the prediction results of related models based on NO. DC00004966 detector's data. Fig 15(b) is a specific part of the comparison results. Fig 16(a) describes the prediction results of state-of-the-art models based on NO. DC00004966 detector's data. Fig 16(b) is a specific part of the comparison results. As shown in Figs 13(a), 14(a), 15(a) and 16(a), it is clearly that the SSA-KELM model have the best fitting performance, especially when the traffic volume changes greatly (it is clearly shown in Figs 13(b), 13(c), 13(d), 14(b), 14(c), 14(d), 15(b) and 16(b)). Fig 17(a) and 17(b) present the frequency of absolute error distribution for all models based on NO. DC00004965 detector's data and NO. DC00004966 detector's data, respectively.

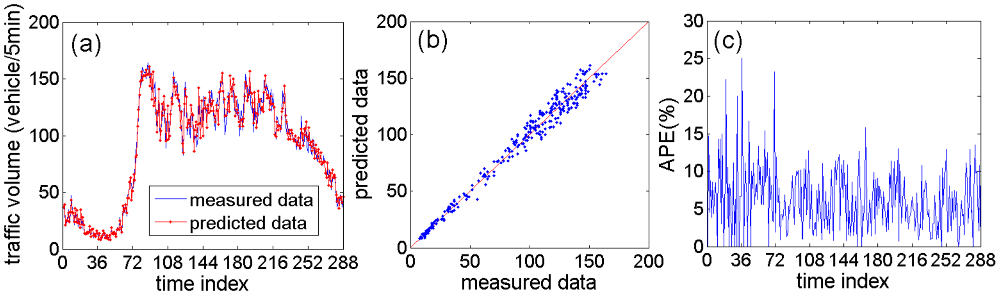

Fig 12. The prediction performance of the proposed method based on NO. DC00004966 detector's data: (a) Curves of predicted data and measured data; (b) Scatterplot of predicted data and measured data; (c) Absolute percent error of predicted data.

doi:10.1371/journal.pone.0161259.g012

Table 2. Parameters of the related models.

Model	parameters
KELM	penalty parameter is 18.5
	RBF kernel parameter is 0.41
SSA-SVM	penalty parameter is 53.6
	RBF kernel parameter is 0.02
SSA-ELM	the number of hidden layer neurons is 27

doi:10.1371/journal.pone.0161259.t002

For the SSA-KELM model, we can clearly see that frequency distribution of absolute errors in the interval [0,5) is the highest, and almost all the absolute errors of SSA-KELM model are less than 25, which illustrates the prediction model has better stability.

Table 3 gives the prediction accuracy by using two statistical indices (MAE and MAPE). The prediction accuracy of the SSA-KELM model is improved significantly compared with the

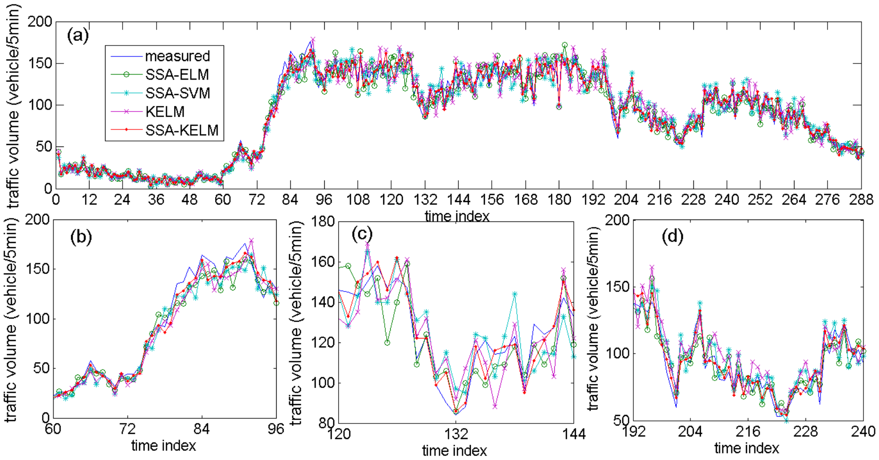

Fig 13. Prediction results of related models based on NO. DC00004965 detector's data (a) and its parts specific prediction results (b), (c), (d).

doi:10.1371/journal.pone.0161259.g013

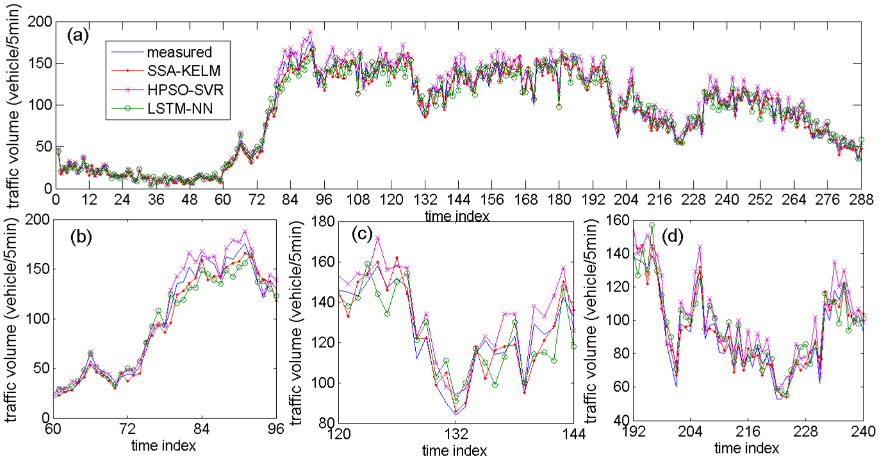

Fig 14. Prediction results of state-of-the-art models based on NO. DC00004965 detector's data (a) and its parts specific prediction results (b), (c), (d).

doi:10.1371/journal.pone.0161259.g014

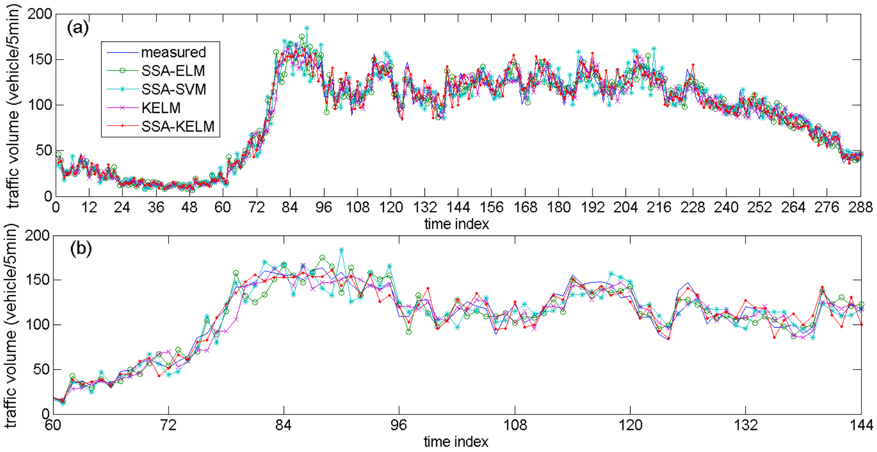

Fig 15. Prediction results of related models based on NO. DC00004966 detector's data (a) and its parts specific prediction results (b).

doi:10.1371/journal.pone.0161259.g015

other three models. More precisely, in the aspect of MAE, the SSA-KELM model gets an extra 28.13% improvement over the SSA-ELM model, an extra 38.53% improvement over the SSA-SVM model, an extra 40.46% improvement over the KELM model, an extra 26.08% improvement over the HPSO-SVR model, and an extra 22.94% improvement over the LSTM-NN model. In the aspect of MAPE, the SSA-KELM model gets an extra 27.55% improvement over the SSA-ELM model, an extra 36.58% improvement over the SSA-SVM model, an extra 38.32% improvement over the KELM model, an extra 29.37% improvement over the HPSO-SVR model, and an extra 24.62% improvement over the LSTM-NN model. The traffic volume time series shown in Figs 13(b), 13(c), 13(d) and 15(b) are denoted as subcase 1, 2, 3 and 4 respectively, because of their great changing during the period (include traffic state transitional period). The performance indices for these four subcases are shown in Table 4, which illustrate that performance of the SSA-KELM model is also better than the other models, when traffic volume changes greatly.

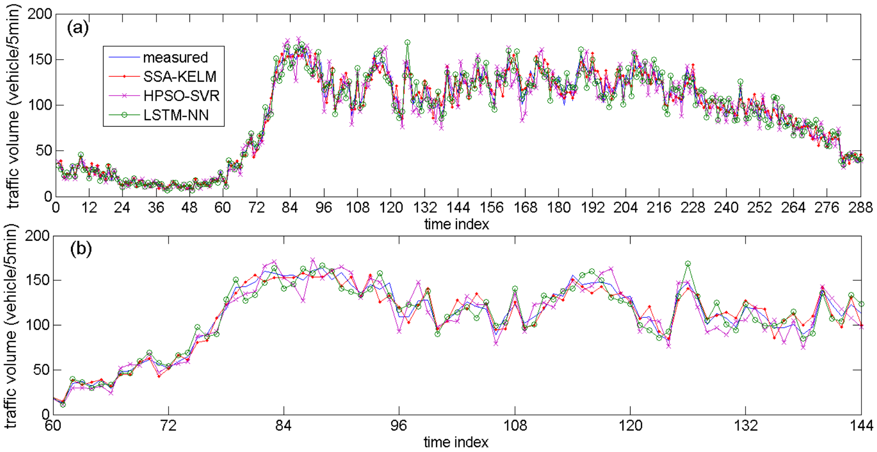

Fig 16. Prediction results of state-of-the-art models based on NO. DC00004966 detector's data (a) and its parts specific prediction results (b).

doi:10.1371/journal.pone.0161259.g016

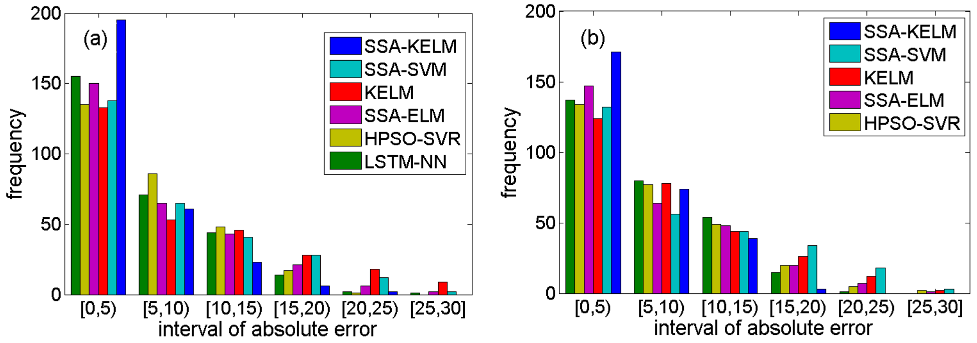

Fig 17. The frequency of absolute error distribution for all models using (a) NO. DC00004965 detector's data and (b) NO. DC00004966 detector's data.

doi:10.1371/journal.pone.0161259.g017

Table 3. Performance indices for different models.

Model	Detector (DC00004965)		Detector (DC00004966)		Mean of the two detectors	
	MAE	MAPE	MAE	MAPE	MAE	MAPE
SSA-ELM	6.78	8.80%	6.93	8.61%	6.86	8.71%
SSA-SVM	7.66	10.00%	8.38	9.89%	8.02	9.95%
KELM	8.64	10.95%	7.92	9.51%	8.28	10.23%
SSA-KELM	4.68	6.44%	5.18	6.18%	4.93	6.31%
HPSO-SVR	6.82	8.98%	7.13	8.36%	6.98	8.67%
LSRM-RNN	6.34	8.54%	6.73	8.02%	6.54	8.28%

doi:10.1371/journal.pone.0161259.t003

Comparison results of the SSA-KELM model, the SSA-ELM model and the SSA-SVM model show that the performance of KELM is better than ELM and SVM in this case. Comparison results of the SSA-KELM model and the KELM model show that using SSA to filter noise of the original traffic volume time series can improve the model's performance effectively. Moreover, the experimental results also demonstrate that the SSA-KELM model achieves good prediction performance using both NO. DC00004965 detector's data and NO. DC00004966 detector's data, which illustrate that SSA-KELM model has a robust generalization ability. Comparison results of the SSA-KELM model, HPSO-SVR model and LSTM-NN model show that the SSA-KELM model can further improve the accuracy of short-term traffic flow prediction.

3.7 Non-parametric tests for multiple comparisons of different models

In order to find significant differences among the results obtained by the studied models, statistical analysis should be performed. In this paper, non-parametric tests are used as suggested in Refs. [59–61]. The parametric statistical analysis loses its credibility because the initial conditions guaranteeing its reliability may not be satisfied [59]. They can offer safe and reliable procedures to contrast the differences between different techniques, especially in multiple-problem analysis.

To compare the performance of multiple models on the multiple data sets, Friedman aligned-ranks test is conducted according to the suggestions of Ref. [60]. Table 5 shows the average ranking of all models. In terms of both MAE and MAPE, the performances of all models can be sorted by average ranking into the following order: SSA-KELM, LSTM-NN,

Table 4. Performance indices of different models for the four subcases.

Model	MAE				MAPE			
	Subcase 1	Subcase 2	Subcase 3	Subcase 4	Subcase 1	Subcase 2	Subcase 3	Subcase 4
SSA-ELM	8.16	9.8	8.04	8.98	8.91%	7.92%	8.92%	8.96%
SSA-SVM	9.92	11.20	8.76	10.84	11.05%	9.35%	10.03%	10.20%
KELM	9.78	11.64	10.69	9.65	10.56%	9.64%	12.17%	9.80%
SSA-KELM	6.30	5.36	5.00	6.59	6.62%	4.37%	5.71%	6.41%
HPSO-SVR	6.86	8.84	8.71	8.65	8.54%	7.92%	9.75%	8.39%
LSTM-RNN	8.78	8.72	7.45	8.08	8.97%	7.34%	8.59%	7.56%

doi:10.1371/journal.pone.0161259.t004

Table 5. Average Rankings of the models using Aligned Friedman.

Model	Ranking (MAE)	Ranking (MAPE)
SSA-KELM	3.5	3.5
KELM	31.3333	31.3333
SSA-SVM	29.6667	29.6667
SSA-ELM	18.3333	18.0833
HPSO-SVR	16	17.5833
LSTM-NN	12.1667	10.8333

doi:10.1371/journal.pone.0161259.t005

HPSO-SVR, SSA-ELM, SSA-SVM and KELM. It means that SSA-KELM and KELM are the best and worst ones among the six models, respectively. Then, if significant differences are found, we will check whether the control model (the best one) significantly outperformed the others (that is, 1×n comparison) using post hoc tests (such as Holm, Hochberg, Hommel and Finner). In this study, we choose the best performing model, SSA-KELM, as the control model for being compared with the rest of models. Under the null hypothesis, the two models are equivalent. If the null hypothesis is rejected, then the performances of these two models are significantly different. In this paper, we only discuss whether the hypotheses are rejected at the 0.05 level of significance. Tables 6 and 7 show the results of adjusted *p* values in terms of MAE and MAPE respectively, where the *p* values below 0.05 are shown in bold. In terms of MAE, Holm, Hochberg and Hommel tests reject hypotheses 1–3, while Finnner test rejects hypotheses 1–4. In terms of MAPE, all the post hoc tests reject hypotheses 1–4. All the results of post hoc tests show that the performance of SSA-KELM is significantly better than that of SSA-ELM, SSA-SVM and EKLM. Partial results of post hoc tests show that the performance of SSA-KELM is significantly better than that of HPSO-SVR. Though SSA-KELM is not statistically better than LSTM-NN, SSA-KELM outperforms LSTM-NN according to the results of the average ranking.

Table 6. Adjusted *p*-values of the models in terms of MAE.

i	Model	unadjusted *p*	p_{Holm}	$p_{Hochberg}$	p_{Hommel}	p_{Finner}
1	KELM	0.000005	**0.000024**	**0.000024**	**0.000024**	**0.000024**
2	SSA-SVM	0.000017	**0.000068**	**0.000068**	**0.000068**	**0.000042**
3	SSA-ELM	0.014745	**0.044235**	**0.044235**	**0.044235**	**0.024454**
4	HPSO-SVR	0.03988	0.07976	0.07976	0.07976	**0.049599**
5	LSTM-NN	0.154218	0.154218	0.154218	0.154218	0.154218

doi:10.1371/journal.pone.0161259.t006

Table 7. Adjusted *p*-values of the models in terms of MAPE.

i	Model	unadjusted *p*	p_{Holm}	$p_{Hochberg}$	p_{Hommel}	p_{Finner}
1	KELM	0.000005	**0.000024**	**0.000024**	**0.000024**	**0.000024**
2	SSA-SVM	0.000017	**0.000068**	**0.000068**	**0.000068**	**0.000042**
3	SSA-ELM	0.016508	**0.049524**	**0.041195**	**0.033016**	**0.027362**
4	HPSO-SVR	0.020597	**0.049524**	**0.041195**	**0.041195**	**0.027362**
5	LSTM-NN	0.227975	0.227975	0.227975	0.227975	0.227975

doi:10.1371/journal.pone.0161259.t007

4 Conclusions

A novel short-term traffic flow prediction model (SSA-KELM) is proposed based on SSA and KELM. The prediction accuracy of SSA-KELM model compares with that of SSA-ELM model, SSA-SVM model and single KELM model using the same real-world traffic volume data. The prediction results are encouraging, especially at the period that traffic volume changes greatly. The main contributions of this paper are not only the introduction of KELM method for traffic flow prediction and how to optimize the model parameters based on GSA, but also considering the chaotic characteristics of short-term traffic flow and determining the model's optimal input form based on PSR. It is worth noting that filtering original traffic volume data in the preprocessing stage is important. The new time series reconstructed by SSA retains the main characteristics of the original traffic volume time series (the contribution ratio is 98.27%), but has filtered the main noise components. A conclusion can be got that the proposed model is an effective and accurate method for short-time traffic flow prediction, which can provide satisfactory prediction results. Although the experimental results presented here are promising and the SSA-KELM model can be successfully applied to predict traffic flow, this model suffers from weak interpretability like other ANN-based models. In future work, the model with well interpretability can be introduced into the proposed model, such as fuzzy system and its improvement [62–64]. It is difficult to improve the interpretability of the model while keeping the accuracy of the model. And it is a difficulty that how to reflect the characteristics of traffic flow in the model.

To obtain more general and robust conclusions, traffic flow data from different roadways and more complicated traffic conditions should be used to test the proposed model. Moreover, it is required that the other data sets of traffic flow parameters (such as travel time, average traffic speed and average occupancy, this study chooses the traffic volume as the demonstration) are applied to test the model. Furthermore, traffic flow parameters data set with different time intervals (such as 1 min, 2 min and 10 min) should be used to test the proposed model for further study. Other advanced optimization algorithms should be further studied to search for more appropriate parameter combinations of the model and to obtain more accurate results of short-term traffic flow prediction.

Supporting Information

S1 Dataset. The traffic volume data from the two detectors.
(XLSX)

Acknowledgments

We thank DX Yu and HX zhou for their help in traffic data collection and analysis.

Author Contributions

Conceptualization: QS CYL ZSY.

Data curation: QS CYL.

Formal analysis: QS CYL.

Funding acquisition: CYL ZSY.

Investigation: QS CYL ZSY QCB XYZ.

Methodology: QS ZSY.

Project administration: CYL.

Resources: QS CYL ZSY.

Software: QS CYL QCB.

Supervision: CYL ZSY.

Validation: QS CYL.

Visualization: QS XYZ.

Writing – original draft: QS.

Writing – review & editing: QS CYL.

References

1. Yang S, Shi S, Hu X, Wang M. Spatiotemporal Context Awareness for Urban Traffic Modeling and Prediction: Sparse Representation Based Variable Selection. PloS one. 2015; 10(10):e0141223. doi: 10.1371/journal.pone.0141223 PMID: 26496370; PubMed Central PMCID: PMC4619804.
2. Castro-Neto M, Jeong Y-S, Jeong M-K, Han LD. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Systems with Applications. 2009; 36(3):6164–73. doi: 10.1016/j.eswa.2008.07.069
3. Guo F, Krishnan R, Polak J. A computationally efficient two-stage method for short-term traffic prediction on urban roads. Transportation Planning and Technology. 2013; 36(1):62–75. doi: 10.1080/03081060.2012.745721
4. Meng M, Shao C-f, Wong Y-d, Wang B-b, Li H-x. A two-stage short-term traffic flow prediction method based on AVL and AKNN techniques. Journal of Central South University. 2015; 22(2):779–86. doi: 10.1007/s11771-015-2582-y
5. Garavello M, Piccoli B. Traffic flow on networks. AITS org, Springfield. 2006.
6. Stathopoulos A, Karlaftis MG. A multivariate state space approach for urban traffic flow modeling and prediction. Transportation Research Part C: Emerging Technologies. 2003; 11(2):121–35. doi: 10.1016/s0968-090x(03)00004-4
7. Laval JA, Leclercq L. Microscopic modeling of the relaxation phenomenon using a macroscopic lane-changing model. Transportation Research Part B: Methodological. 2008; 42(6):511–22. doi: 10.1016/j.trb.2007.10.004
8. Gipps P G. A behavioural car-following model for computer simulation. Transportation Research Part B: Methodological. 1981; 15(2): 105–111. doi: 10.1016/0191-2615(81)90037-0
9. Sun H, Liu H X, Xiao H, He R R, Ran B. Short-term traffic forecasting using the local linear regression model. Transportation Research Record: Journal of the Transportation Research Board. 2003; 1836: 143–150. http://dx.doi.org/10.3141/1836-18.
10. Williams B M, Hoel L A. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. Journal of transportation engineering. 2003; 129(6): 664–672.
11. Lint H V, Hoogendoorn S P, Zuylen H V. Freeway travel time prediction with state-space neural networks: Modeling state-space dynamics with recurrent neural networks. Transportation Research Record: Journal of the Transportation Research Board. 2002; 1811: 30–39. http://dx.doi.org/10.3141/1811-04.

12. Jiang X, Adeli H. Dynamic wavelet neural network model for traffic flow forecasting. Journal of transportation engineering. 2005; 131(10): 771–779.

13. Chandra SR, Al-Deek H. Predictions of Freeway Traffic Speeds and Volumes Using Vector Autoregressive Models. Journal of Intelligent Transportation Systems. 2009; 13(2):53–72. doi: 10.1080/15472450902858368

14. Wu C H, Ho J M, Lee D T. Travel-time prediction with support vector regression. Intelligent Transportation Systems, IEEE Transactions on. 2004; 5(4): 276–281. doi: 10.1109/TITS.2004.837813

15. Vlahogianni EI, Karlaftis MG, Golias JC. Short-term traffic forecasting: Where we are and where we're going. Transportation Research Part C: Emerging Technologies. 2014; 43:3–19. doi: 10.1016/j.trc.2014.01.005

16. Vlahogianni EI, Golias JC, Karlaftis MG. Short-term traffic forecasting: Overview of objectives and methods. Transport Reviews. 2004; 24(5):533–57. doi: 10.1080/0144164042000195072

17. Van Lint J W C, Van Hinsbergen C. Short-term traffic and travel time prediction models. Artificial Intelligence Applications to Critical Transportation Issues. 2012; 22: 22–41.

18. Huang G-B, Zhu Q-Y, Siew C-K. Extreme learning machine: Theory and applications. Neurocomputing. 2006; 70(1–3):489–501. doi: 10.1016/j.neucom.2005.12.126

19. Yu Q, Miche Y, Séverin E, Lendasse A. Bankruptcy prediction using extreme learning machine and financial expertise. Neurocomputing. 2014; 128: 296–302. doi: 10.1016/j.neucom.2013.01.063

20. Wang D D, Wang R, Yan H. Fast prediction of protein-protein interaction sites based on Extreme Learning Machines. Neurocomputing. 2014; 128: 258–266. doi: 10.1016/j.neucom.2012.12.062

21. Ghouti L, Sheltami T R, Alutaibi K S. Mobility prediction in mobile ad hoc networks using extreme learning machines. Procedia Computer Science. 2013; 19: 305–312. doi: 10.1016/j.procs.2013.06.043

22. Deng Z, Choi K S, Cao L, Wang S. T2fela: type-2 fuzzy extreme learning algorithm for fast training of interval type-2 TSK fuzzy logic system. IEEE transactions on neural networks and learning systems. 2014; 25(4): 664–676. doi: 10.1109/TNNLS.2013.2280171 PMID: 24807945

23. He Y L, Geng Z Q, Zhu Q X. A data-attribute-space-oriented double parallel (DASODP) structure for enhancing extreme learning machine: Applications to regression datasets. Engineering Applications of Artificial Intelligence. 2015; 41: 65–74. doi: 10.1016/j.engappai.2015.02.001

24. Bencherif M A, Bazi Y, Guessoum A, Alajlan N, Melgani F, AlHichri H. Fusion of extreme learning machine and graph-based optimization methods for active classification of remote sensing images. IEEE Geoscience and Remote Sensing Letters. 2015; 12(3): 527–531. doi: 10.1109/LGRS.2014.2349538

25. Wang S, Jiang Y, Chung F L, Qian P. Feedforward kernel neural networks, generalized least learning machine, and its deep learning with application to image classification. Applied Soft Computing. 2015; 37: 125–141. doi: 10.1016/j.asoc.2015.07.040

26. Huang G B, Zhou H, Ding X, Zhang R. Extreme learning machine for regression and multiclass classification. IEEE Trans. Syste. Man Cybern. B: Cybern. 2012; 42 (2): 513–529. doi: 10.1109/TSMCB.2011.2168604

27. Wang X, Han X. Online sequential extreme learning machine with kernels for non stationary time series prediction. Neurocomputing. 2014; 145: 90–97. doi: 10.1016/j.neucom.2014.05.068

28. Shamshirband S, Mohammadi K, Chen H L, Samy G N, Petković D, Ma C. Daily global solar radiation prediction from air temperatures using kernel extreme learning machine: A case study for Iran. Journal of Atmospheric and Solar-Terrestrial Physics.2015; 134: 109–117. doi: 10.1016/j.jastp.2015.09.014

29. Huang G B. An insight into extreme learning machines: random neurons, random features and kernels. Cognitive Computation. 2014; 6(3): 376–390. doi: 10.1007/s12559-014-9255-2

30. Chen H L, Wang G, Ma C, Cai Z N, Liu W B, Wang S J. An efficient hybrid kernel extreme learning machine approach for early diagnosis of Parkinson′ s disease. Neurocomputing. 2016; 186: 131–144. doi: 10.1016/j.neucom.2015.07.138

31. Rashedi E, Pour HN, Saryazdi S. GSA: a gravitational search algorithm. Inform Sci 2009; 179: 2232–2248. doi: 10.1016/j.ins.2009.03.004

32. Li C, An X, Li R. A chaos embedded GSA-SVM hybrid system for classification. Neural Comput. Appl. 2015; 26 (3), 713–721. doi: 10.1007/s00521-014-1757-z

33. Yuan X, Chen C, Yuan Y, Huang Y, Tan Q. Short-term wind power prediction based on LSSVM-GSA model. Energy Conversion and Management. 2015; 101: 393–401. doi: 10.1016/j.enconman.2015.05.065

34. Zhang W, Niu P, Li G, Li P. Forecasting of turbine heat rate with online least squares support vector machine based on gravitational search algorithm. Knowledge-Based Systems. 2013; 39: 34–44. doi: 10.1016/j.knosys.2012.10.004

35. Dendrinos D S. Traffic-flow dynamics: a search for chaos. Chaos, Solitons & Fractals. 1994; 4(4): 605–617. doi: 10.1016/0960-0779(94)90069-8

36. Lawrence W, April Y, Lin F Y. Testing and prediction of traffic flow dynamics with chaos. Journal of the Eastern Asia Society for Transportation Studies. 2003; 5: 1975–1990.

37. Frazier C, Kocklman K. Chaos theory and transportation systems: instructive example. Journal of Transportation Research Board. 2004; 1897: 9–17. http://dx.doi.org/10.3141/1897-02.

38. Holzfuss J, Mayer-Kress G. An approach to error estimation in the application of dimension algorithms. Dimensions and Entropies in Chaotic Systems. 1986; 32: 114–122. doi: 10.1007/978-3-642-71001-8_15

39. Fraser A M, Swinney H L. Independent coordinates for strange attractors from mutual information. Physical review A. 1986; 33(2): 1134. http://dx.doi.org/10.1103/PhysRevA.33.1134

40. Rosenstein M T, Collins J J, De Luca C J. Reconstruction expansion as a geometry-based framework for choosing proper delay times. Physica D: Nonlinear Phenomena. 1994; 73(1): 82–98. doi: 10.1016/0167-2789(94)90226-7 10.1016/0167-2789(94)90226-7.

41. Grassberger P, Procaccia I. Characterization of strange attractors. Physical Review Letters. 1983; 50 (5): 346–349. http://dx.doi.org/10.1103/PhysRevLett.50.346.

42. Kennel M B, Brown R, Abarbanel H D I. Determining embedding dimension for phase-space reconstruction using a geometrical construction. Physical review A. 1992; 45(6): 3403. http://dx.doi.org/10.1103/PhysRevA.45.3403.

43. Cao L. Practical method for determining the minimum embedding dimension of a scalar time series. Physica D: Nonlinear Phenomena. 1997; 110(1–2): 43–50. doi: 10.1016/S0167-2789(97)00118-8

44. Kim H S, Eykholt R, Salas J D. Nonlinear dynamics, delay times, and embedding windows. Physica D: Nonlinear Phenomena. 1999; 127(1): 48–60. doi: 10.1016/S0167-2789(98)00240-1

45. Soltani S. On the use of the wavelet decomposition for time series prediction. Neurocomputing. 2002; 48(1): 267–277. doi: 10.1016/S0925-2312(01)00648-8

46. Hassani H, Mahmoudvand R. Multivariate singular spectrum analysis: A general view and new vector forecasting approach. International Journal of Energy and Statistics. 2013; 1(01): 55–83. doi: 10.1142/S2335680413500051

47. Abdollahzade M, Miranian A, Hassani H, Hossein I. A new hybrid enhanced local linear neuro-fuzzy model based on the optimized singular spectrum analysis and its application for nonlinear and chaotic time series forecasting. Information Sciences. 2015; 295: 107–125. doi: 10.1016/j.ins.2014.09.002

48. Hassani H, Soofi A S, Zhigljavsky A A. Predicting daily exchange rate with singular spectrum analysis. Nonlinear Analysis: Real World Applications. 2010; 11(3): 2023–2034. doi: 10.1016/j.nonrwa.2009.05.008

49. Afshar K, Bigdeli N. Data analysis and short term load forecasting in Iran electricity market using singular spectral analysis (SSA). Energy. 2011; 36(5): 2620–2627. doi: 10.1016/j.energy.2011.02.003

50. Zhang Q, Wang B D, He B, Peng Y, Ren M L. Singular spectrum analysis and ARIMA hybrid model for annual runoff forecasting. Water resources management. 2011; 25(11): 2683–2703. doi: 10.1007/s11269-011-9833-y

51. Sanei S, Ghodsi M, Hassani H. An adaptive singular spectrum analysis approach to murmur detection from heart sounds. Medical engineering & physics. 2011; 33(3): 362–367.

52. Golyandina N, Nekrutkin V, Zhigljavsky A A. Analysis of Time Series Structure: SSA and Related Techniques. CRC Press, New York. 2001.

53. Takens F. Detecting strange attractors in turbulence. Lecture Notes in Mathematics, Springer-Verlag. 1981; 898: 366–381. doi: 10.1007/BFb0091924

54. Brock W A, Hsieh D A, LeBaron B D. Nonlinear dynamics, chaos, and instability: statistical theory and economic evidence. MIT press, Cambridge. 1991.

55. Hassani H, Heravi S, Zhigljavsky A. Forecasting European industrial production with singular spectrum analysis. International journal of forecasting. 2009; 25(1): 103–118. doi: 10.1016/j.ijforecast.2008.09.007

56. Zhang Y, Yang Y. Cross-validation for selecting a model selection procedure. Journal of Econometrics. 2015; 187(1): 95–112. doi: 10.1016/j.jeconom.2015.02.006

57. Hu W, Yan L, Liu K, Wang H. A short-term traffic flow forecasting method based on the hybrid PSO-SVR. Neural Processing Letters. 2016; 43(1): 155–172. doi: 10.1007/s11063-015-9409-6

58. Ma X, Tao Z, Wang Y, Yu H, Wang Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transportation Research Part C: Emerging Technologies. 2015; 54: 187–197. doi: 10.1016/j.trc.2015.03.014

59. Demšar J. Statistical comparisons of classifiers over multiple data sets. Journal of Machine learning research. 2006; 7(Jan): 1–30.

60. García S, Fernández A, Luengo J, Herrera F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences. 2010; 180(10): 2044–2064. doi: 10.1016/j.ins.2009.12.010

61. Derrac J, García S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation. 2011: 1(1): 3–18. doi: 10.1016/j.swevo.2011.02.002

62. Jiang Y, Chung F L, Ishibuchi H, Deng Z, Wang S. Multitask TSK fuzzy system modeling by mining intertask common hidden structure. IEEE transactions on cybernetics. 2015; 45(3): 534–547. doi: 10.1109/TCYB.2014.2330844

63. Deng Z, Choi K S, Jiang Y, Wang S. Generalized hidden-mapping ridge regression, knowledge-leveraged inductive transfer learning for neural networks, fuzzy systems and kernel methods. IEEE transactions on cybernetics. 2015; 44(12): 2585–2599. doi: 10.1109/TCYB.2014.2311014

64. Olatunji S O, Selamat A, Abdulraheem A. A hybrid model through the fusion of type-2 fuzzy logic systems and extreme learning machines for modelling permeability prediction. Information fusion. 2014; 16: 29–45. doi: 10.1016/j.inffus.2012.06.001