Short-term Traffic Flow Prediction Based on KELM Optimized By Improved Slime Mould Algorithm

Ming Zhao[*], *Jian Tang, Guo-Qing Chang*

School of Air Transportation, Shanghai University of Engineering Science, China
zhaoming@sues.edu.cn, 1289941640@qq.com, 1012983428@qq.com

Abstract

Traffic flow prediction plays a crucial role in improving transportation efficiency and enhancing Intelligent Transportation Systems (ITS). However, the temporal, spatial, and nonlinear nature of traffic flow data presents challenges for accurate short-term prediction. We propose a short-term traffic flow prediction model based on Kernel Extreme Learning Machine (KELM) optimized by the improved Slime Mould Algorithm (SMA). KELM is an improved version of Extreme Learning Machine (ELM) that incorporates kernel functions for improved generalization and stability. SMA is a meta-heuristic algorithm inspired by the behavior of slime mould in foraging, known for its strong global searching ability. For better performance, three strategies are introduced: the Good Point Set method for optimizing the initial population, the combination of Opposition Based Learning (OBL) and Differential Evolution (DE) to improve the slime mould generation mechanism, and the use of adaptive t distribution mutation to enhance convergence speed. After comparing the performance of these improved SMAs on twelve test functions, the ISMA improved by integrating three strategies as mentioned above is best. Then the ISMA is applied to search for the optimal parameters of KELM model. Finally, the optimized KELM with optimal parameters is applied to predict the short-term traffic flow on given traffic data set. Experimental results demonstrate that the proposed model, KELM optimized by ISMA namely ISMA-KELM, outperforms existing models such as Random Forest (RF), Least Squares Support Vector Machine (LSSVM), KELM optimized by Tuna Swarm Optimization Algorithm (TSO-KELM), and KELM optimized by SMA (SMA-KELM) in terms of traffic flow prediction accuracy. The proposed model ISMA-KELM provides a promising approach for addressing the challenges of traffic flow prediction, offering improved accuracy and efficiency in real-time traffic management systems.

Keywords: Short-term traffic flow, Kernel extreme learning machine, Slime mould algorithm, Prediction, Optimization

1 Introduction

The objective of traffic flow prediction is to deliver precise and real-time traffic flow information within the next 10-15 minutes to enhance transportation efficiency in the field of ITSs [1-2]. The traffic flow data exhibit temporal, spatial and nonlinear characteristics, which brings challenges and difficulties for achieving effective and accurate short-term traffic flow prediction.

Recent research on traffic flow prediction can be categorized as follows: (i) prediction based on linear theories, including History Average Model, Kalman Filtering Model, Time-Series Model, and others; (ii) prediction based on nonlinear theories, including wavelet analysis, chaos theory and catastrophe theory [3]; (iii) prediction based on simulation model, such as combining with the classical model VISSIM to achieve road network simulation collaborative control [4]; (iv) prediction based on intelligent theory, including neural network, support vector machine, and deep learning; (v) prediction based on hybrid models, where some utilize multiple models for cohesion and cooperation to obtain the final result, while others involve simultaneous predictions by several models, followed by the fusion of results using a specific strategy [5].

Neural network has been demonstrated effective prediction capabilities in traffic flow prediction [6]. However, the selection of parameters lacks clarity and a more suitable method, often relying on individual subjective repeated experiments, leading to potential issues with local optimality and overfitting. ELM is an enhanced feedforward neural network featuring a single hidden layer. The algorithm randomly assigns weights and thresholds to the input and hidden layers, allowing for the determination of a unique optimal solution by specifying the number of neurons in the input and hidden layer without resetting values during training. Consequently, ELM offers the advantage of rapid learning. A model for urban traffic flow prediction based on ELM neural network was proposed in [7]. Comparative analysis with the traditional BP neural network demonstrated superior prediction accuracy, convergence, rapid learning speed, and good generalization performance of the proposed model. Since the input weight and implicit bias of ELM can be randomly determined during training, suboptimal input weight and bias may arise, whereas the output weight is calculated based on these input values. To address this issue, Sun Tao et al. proposed the utilization of Differential Evolution (DE) to optimize the weight of ELM, thereby achieving a global optimal solution and enhancing the overall generalization performance of the algorithm [8]. Considering the uncertain changes in traffic flow, Wang Tao

[*]Corresponding Author: Ming Zhao; E-mail: zhaoming@sues.edu.cn

DOI: https://doi.org/10.70003/160792642024092505001

et al. proposed the real-time update parameters of prediction model using ELM with a forgetting factor [9]. Additionally, they employed particle filtering to mitigate random noise in the system, thereby achieving optimal short-time change estimation of traffic volume and enhancing the real-time prediction accuracy of traffic flow. To mitigate the impact of randomness in the input weight matrix and hidden layer bias on ELM, Xiangpeng Cui et al. suggested the utilization of Genetic Algorithm (GA) to optimize the components, thereby achieving superior prediction accuracy and efficiency [10]. Given that traffic flow data exhibit high temporal variability, nonlinearity, and susceptibility to random noise, Ruqing Chen et al. adopted singular spectrum analysis to remove noise from the original data [11]. Furthermore, they optimized the ELM weight threshold through a combination of the fireworks algorithm and the difference algorithm. Considering that traffic flow data are continuously updated in real-time, the prediction model requires rebuilding whenever the historical dataset is updated. However, this process consumes significant computing resources and running time. To achieve dynamic online updating of the traffic flow prediction model, Borong Zhou et al. proposed the integration of Auto-Encoder (AE) within ELM to learn the weight and bias parameters of the input and hidden layers [12]. Additionally, Recurrent Neural Network (RNN) was employed to extract temporal variation features of traffic flow data. The prediction time of models is short, which ensures a certain real-time performance.

In practical applications, ELM encounters challenges such as the difficulty in determining model parameters, poor robustness, and a tendency to overfit. So the kernel function theory from support vector machine was adopted, resulting in an enhanced version named Kernel Extreme Learning Machine (KELM). This modification aimed to improve stability, classification robustness, and prediction accuracy. Meanwhile, the regularization coefficient and kernel parameters can affect the performance of KELM model. Therefore, various methods have been applied for optimizing parameter selection or improving the performance of KELM model [13]. Changsheng Zhu et al. addressed the issue of low wind power prediction accuracy due to the volatility and non-stationarity of wind energy by proposing a wind power prediction model based on variable mode decomposition technology and an improved grey wolf algorithm to optimize KELM [14]. In the basic grey wolf algorithm, two types of target search exist: local search and global search. To enhance the optimization capability of grey wolf algorithm and mitigate its susceptibility to local optimal solution, a differential optimization algorithm is incorporated. And a nonlinear convergence factor is used to balance local and global searching. Then, this improved Gray Wolf algorithm (DIGWO) is utilized to improve the KELM prediction model. For further improving the prediction accuracy under the limited monitoring data, Feiyan Ma et al. utilized sparrow search algorithm to optimize the regularization coefficient and kernel function parameters of KELM model, and established a landslide displacement prediction model to improve the prediction accuracy [15]. Due to its real-time performance, good classification accuracy and stability, KELM has a good advantage in traffic state discrimination. Huiru Chen et al. built a traffic state discrimination decision model based on KELM improved by GA [16]. A bee colony optimization KELM model was proposed in [17], the correct rate of recognition and classification has increased by nearly 20%, while the misjudgment rate has dropped to 3%. To improve the simulation value of a single prediction model, a nonlinear combined prediction model with time weights was proposed in [18], and the variable weights were generated by KELM which used the radial basis function to assign the output matrix parameters.

Considering the influence of the regularization coefficient and kernel function parameters to the KLEM model, this paper proposes a KLEM model optimized by improved SMA, then predict the short-term traffic flow. SMA is known for its strong global searching ability in optimizing at any direction and any step size, but its convergent speed and precision could be improved further [19]. Therefore, in this paper, the SMA is improved by three strategies, such as searching for excellent initial group by good point set method, optimizing slime mould formation mechanism by Opposition Based Learning and Differential Evolution, and increasing the convergence speed by adaptive *t* distribution mutation. During experimental stage, three strategies are applied step by step. Firstly, the SMA is enhanced by good point set method renamed as GSMA, and the SMA is enhanced by combining good point set method and opposition differential evolution renamed as GODSMA, then the SMA is enhanced by combining good point set method, opposition differential evolution and adaptive t distribution mutation renamed as ISMA. Secondly, comparing the performance of SMA, GMSA, GODSMA and ISMA on twelve test functions, the best improved SMA is selected, which is applied to optimize the parameters of KELM. On given traffic flow data set, the experimental results show that the KELM optimized by ISMA has better prediction accuracy than any other existing model such as Random Forest, Least Squares Support Vector Machine, KELM optimized by Tuna Swarm Optimization Algorithm, and KELM optimized by standard SMA.

2 Methodology

The traffic system is a highly complex nonlinear dynamic system. Traffic flow data not only exhibits temporal proximity and spatial correlation, but also is susceptive to external factors such as road conditions, weather, and monitoring equipment. Deep neural networks excel at dealing with complex and nonlinear problems, however, the parameter learning process is relatively tedious. ELM possesses attributes such as simplicity, effectiveness, a limited number of training parameters, and strong generalization capabilities. Then KELM is an improvement of ELM that incorporates a kernel function to enhance the generalization and stability of the neural network. Moreover, KELM inherits all the advantages of ELM. Consequently, this paper proposed a short-term traffic flow prediction model based on KELM by utilizing the improved SMA for parameter optimization and input structure refinement of KELM.

2.1 Kernel Extreme Learning Machine

ELM generates random continuous weight value *w* between the input and hidden layers, and the bias *b* of the hidden layer neurons. In the training process, the optimal solution requires manual specification of the number of hidden layer neurons *l*. For Q different samples $(x_i, t_i), i = 1, 2, \cdots, Q$, let *β* represent the weight matrix of the output layer and *H* represent the hidden layer output matrix. Thus, *T* can be expressed as *Hβ* = *T*, indicating the expected output matrix. The output weights *β* can be derived by solving the least square solution of the following equations, denoted as $\beta = H^{+}T$, where H^{+} representing the Moore-Penrose generalized inverse of matrix *H*.

Since the random generation of input weight and implicit neuron bias in ELM makes it challenging to determine the model parameters, leading to a susceptibility to local optima [20]. The number of neurons in hidden layer must be manually specified, while previous research has demonstrated that increasing the number of nodes does not always yield superior results [21]. The correlation between the optimal accuracy of diverse datasets and the number of hidden layer nodes is intricate.

Considering the limitations of ELM, the kernel function was incorporated into the model, resulting in the KELM [22]. For enhancing the stability of neural network, the regularization coefficient C is introduced using orthogonal projection and ridge regression theory, resulting in the formulation of the output weights as:

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

The kernel function replaces the random feature mapping in ELM, and the resulting kernel matrix can be represented as:

$$\Omega_{ELM} = HH^{T} = h(x_i)h(x_j) = K(x_i, x_j) \tag{2}$$

Consequently, the output function is defined as:

$$\begin{aligned} F(x) &= h(x)\beta = h(x)H^{T}\left(\frac{I}{C} + HH^{T}\right)^{-1} T \\ &= \begin{bmatrix} K(x, x_1) \\ K(x, x_2) \\ \vdots \\ K(x, x_n) \end{bmatrix}^{T} \left(\frac{I}{C} + \Omega_{ELM}\right)^{-1} T \end{aligned} \tag{3}$$

where *I* represents the unit matrix, (x_1, x_2, …, x_n) signifies the given training sample, *n* denotes the number of samples, and *K* (x_i, x_j) denotes the kernel function.

Despite addressing the random initialization issue of ELM and possessing fewer adjustable parameters, faster convergence speed, and improved generalization performance, the performance of KELM is influenced by the regularization coefficient *C* and kernel function, as observed in Equation (3). Therefore, in this study, we propose to optimize the parameters of KELM using SMA due to its strong global searching capabilities, with the aim of enhancing traffic flow prediction accuracy.

2.2 Standard Slime Mould Algorithm

SMA is a swarm intelligence algorithm that simulates the positive and negative feedback processes generated by the weight's changes of slime mould during foraging [23]. It can perform global optimization in any direction and at any step size.

In order to model the food approach behavior of slime mould, the following rule is proposed.

$$\overrightarrow{X(t+1)} = \begin{cases} \overrightarrow{X_b(t)} + \overrightarrow{vb} \cdot \left(\overrightarrow{W} \cdot \overrightarrow{X_A(t)} - \overrightarrow{X_B(t)}\right), r < p \\ \overrightarrow{vc} \cdot \overrightarrow{X(t)}, r \geq p \end{cases} \tag{4}$$

where $\overrightarrow{X}$ represents the position of slime mould, $\overrightarrow{X_b}$ represents the individual position with the highest current odor concentration, $\overrightarrow{X_A}$ and $\overrightarrow{X_B}$ represent two randomly selected individuals from the slime mould, $\overrightarrow{W}$ denotes the weight of the slime mould, $\overrightarrow{vb}$ is a parameter with a range of [- a, a], and $\overrightarrow{vc}$ linearly decreases from 1 to 0 with each iteration, *t* denotes the current iteration [23-24], *r* represents a random value in the range [0,1]. The formula for *p* is as follows:

$$p = \tanh\left|S(i) - DF\right| \tag{5}$$

where *S*(*i*) represents the fitness of slime mould $\overrightarrow{X}$, $i \in 1$, 2, …, *N*. And *DF* represents the best fitness obtained in all iterations.

The position update of slime mould can be represented as follows:

$$\overrightarrow{X^{*}} = \begin{cases} rand \cdot (UB - LB) + LB, rand < z \\ \overrightarrow{X_b(t)} + \overrightarrow{vb} \cdot \left(\overrightarrow{W} \cdot \overrightarrow{X_A(t)} - \overrightarrow{X_B(t)}\right), r < p \\ \overrightarrow{vc} \cdot \overrightarrow{X(t)}, r \geq p \end{cases} \tag{6}$$

where LB and UB represent the upper and lower limits of the search range, while rand and r represent random values in the range [0,1]. z is a parameter that balances the search and development stages, and its typical empirical value is 0.03.

The $\overrightarrow{vb}$ value randomly oscillates within the range [-a, a] and gradually approaches zero with each iteration. It oscillates between [- 1,1] and eventually converges to zero as follows:

$$\overrightarrow{vc} = \left[-\left(1 - \frac{t}{\max\_t}\right), 1 - \frac{t}{\max\_t}\right] \tag{7}$$

Previous studies have employed SMA to optimize the penalty parameters and kernel parameters of LSSVM, and the results demonstrate a lower mean square error for LSSVM improved by SMA [25]. Chuantao Zang et al. also utilized SMA to optimize the learning rate, training times and the number of hidden layer neurons in LSTM for predicting the remaining life of bearings [26].

Due to randomly generating the initial population of SMA without supporting of prior knowledge, the initial population quality is poor, which hampers global optimization. Furthermore, using the last position of the previous iteration as the initial position for each iteration diminishes the diversity of slime moulds and reduces convergence accuracy. Moreover, the late iteration oscillation effect of the algorithm is weak, making it prone to getting trapped in local optima. In practical applications, several methods are employed to enhance the global search ability, improve the convergence accuracy and speed of the algorithm. Tent chaotic mapping was introduced to increase population diversity, and the elite opposition-based learning strategy was developed to expand the search range. These approaches enhance the convergence speed and solution precision of the standard slime mould algorithm [27]. Since slime molds that go beyond the boundary are considerably distant from the optimal fitness position, making it challenging for them to return to the optimal position. As a result, the algorithm's convergence speed decreases. To address this, boundary conditions tailored to the characteristics of photovoltaic array output are proposed to reduce the number of iterations of slime molds beyond the boundary and improve the algorithm's convergence speed. Additionally, the Levy flight strategy is introduced to optimize convergence rules and improve the random search ability, enabling faster tracking towards the global optimum [19].

3 Proposed Improvement of SMA

This paper introduces three strategies to enhance the search ability of SMA, namely, Good Point Set for optimizing the initial population, the combination of Opposition Based Learning and Differential Evolution to improve the slime mould generation mechanism, and the utilization of adaptive *t* distribution mutation to improve the convergence speed.

3.1 Optimizing Initial Population

Due to the randomness of slime mould, the initial population quality is low. The Good Point Set strategy is introduced to initialize the population, ensuring an even distribution of the initial slime mould population in the search space, thereby improving the optimization effect of SMA.

Let G_d represent the unit cube in a d dimensional Euclidean space. If $r \in G_d$. The deviation for $P_n(k)$= {({$r_1^{(n)}\cdot k$}, {$r_2^{(n)}\cdot k$}, ..., {$r_d^{(n)}\cdot k$}), $1 \le k \le n$} is denoted by $\varphi(n)=C(r,\varepsilon)n^{-1+\varepsilon}$, and $C(r,\varepsilon)$ is a constant only related to r, $\varepsilon(\varepsilon>0)$, then $P_n(k)$ is referred to the set of good points, with r representing the best point. And {$r_d^{(n)}\cdot k$} represents the decimal part of $r_d^{(n)}\cdot k$, n stands for points. In this paper, r_k = {$2\cos(\frac{2\pi k}{p})$, $1 \le k \le d$}, p represents the minimum prime number satisfying the given condition $\frac{p-3}{2} \ge d$ [28].

The initialization processing using the Good Point Set is as follows:

(1) Construct a set of good points containing points X = {$x_1, x_2, ... x_i, ..., x_N$} ($i$ = 1, 2, ..., N), where N represents the population size.

(2) Assign $x_{ij} = i*2\cos(\frac{2\pi j}{p})$ to any dimensional component in $x_i = (x_{i1}, x_{i2}, ...x_{ij}, x_{iD})$ (j = 1, 2, ..., D), where D represents the dimension of search space.

(3) Map the set of good points to the search space of the optimization problem using the specific calculation formula $x'_{ij} = LB_j + mod\,(x_{ij}, 1) * (UB_j - LB_j)$, Where x'_{ij} represents the position of the slime mould, and represent the upper and LB_j, UB_j lower bounds of the corresponding dimensions in the search space.

3.2 Improving Slime Mould Formation Mechanism

This paper proposes an improved slime mould generation mechanism that combines Opposition Based Learning (OBL) and Differential Evolution (DE). The mechanism evaluates the reverse solution of the current solution and selects the optimal solution as the next generation individual, while the diversity of population is enhanced by DE to improve the global optimization performance and convergence accuracy of the algorithm.

Let $x \in R$ be a real number defined within a specific interval, denoted as $x \in [lb, ub]$, where lb and ub represent the upper and lower bounds of the problem, respectively. The opposite number of $\tilde{x}$ is defined as follows:

$$\tilde{x} = lb + ub - x \tag{8}$$

Similarly, this concept can be extended to multidimensional space. Let $x = (x_1, x_2, ..., x_D)$ be a point in the D coordinate system, denoted as $(x_1, x_2, ..., x_D)$, $x_i \in [lb_i, ub_i]$. The opposite point of $\tilde{x}$ is defined by its coordinates as $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_D$.

$$x_i = lb_i + ub_i - x_i, i = 1, 2, ..., D \tag{9}$$

Therefore, in the first stage of the proposed algorithm, the reverse population is calculated using Equation (10) from N original populations. Subsequently, a population of $2N$ slime moulds is created, consisting of positive slime moulds and their respective reverse counterparts, which are used in the DE operation.

DE is essentially an improved genetic algorithm that primarily includes mutation, crossover, and selection strategies to find the optimal solution to a problem. Firstly, a mutation operation is performed on the slime mould population as shown in Equation (10).

$$v_i^{t+1} = X_{r_1}^t + R\left(X_{r_2}^t - X_{r_3}^t\right) \tag{10}$$

r_1, r_2, $r_3 \in [1, 2N]$ represents a different random number, while R denotes the scaling factor for mutation and takes a random number in [0,1], and t is the number of iterations.

Next, the newly generated slime mould resulting from the variation is hybridized with the original slime mould using the following formula:

$$u_{i,j}^{t} = \begin{cases} v_{i,j}^{t+1}, \ if \ rand \le P_c \\ X_{i,j}^{t}, \ otherwise \end{cases} \tag{11}$$

In the above formula, *rand* represents a random number in the range [0,1], while P_c denotes the crossover probability with a value of P_c=0.5.

Finally, a choice is made based on the following condition:

$$X_i^{t+1} = \begin{cases} u_i^t, if \ f(u_i^t) \le f(X_i^t) \\ X_i^t, \quad otherwise \end{cases} \tag{12}$$

where $f(X_i^t)$ represents the fitness value of the original slime mould, and $f(u_i^t)$ represents the fitness value of the new slime mould generated by crossover operation.

3.3 Improving Convergence Speed

The shape of the *t*-distribution curve depends on the degree of freedom parameter, denoted as *n*. The higher degree of freedom *n* results in a taller curve shape, while a lower degree of freedom *n* results in a flatter curve shape, specifically resembling $t(n \to \infty) \to N(0,1)$, $t(n \to 1) \to C(0,1)$. Here, $N(0,1)$ represents the Gaussian distribution and $C(0,1)$ represents the Cauchy distribution. Thus, the standard Gaussian distribution and Cauchy distribution are two special cases at the boundaries of the *t*-distribution.

During the iteration process, the position of the slime mould is adaptively varied according to the *t*-distribution based on the average fitness of the current iteration. This strategy utilizes the information interference from the current population to enable the slime mould to escape local optimum, converge towards the global optimum, and improve the convergence speed of the algorithm.

The position state of slime mould, denoted as $X_i = (X_{i1}, X_{i2}, \ldots, X_{iD})$, is defined by the following formula:

$$\overrightarrow{X(t)} = \begin{cases} X_i + X_i * t(iter), f(X_i) < f(ave) \\ (X_i + X_r)/2, f(X_i) \ge f(ave) \end{cases} \tag{13}$$

where $\overrightarrow{X(t)}$ represents the individual position of the mutant slime mould, X_i represents the current location of the slime mould, $f(X_i)$ represents the fitness of the slime mould, X_r represents the randomly selected individual position of the myxomycetes in the top half of the fitness ranking, $f(ave)$ represents the average fitness of all populations, and $t(iter)$ represents the t-distribution with the number of algorithm iterations denoted as *iter* as the parameter freedom. In the early stages of iteration, when the number of iterations is small, the variation *t*-distribution is similar to that of the Cauchy distribution, providing the algorithm with a strong capability for global exploration. Conversely, in the later stages of iteration, when the number of iterations is large, the *t*-distribution variation resembles that of the Gaussian distribution, enhancing the algorithm's capacity for local development and accelerating convergence speed.

3.4 Evaluation of Improved SMA

After introducing three strategies to standard SMA, we have renamed the improved slime mould algorithm with Good Point Set as GSMA, the improved slime mould algorithm with Good Point Set and Oppositional Differential Evolution as GODSMA, and the improved slime mould algorithm with Good Point Set, Oppositional Differential Evolution and adaptive *t*-distribution mutation as ISMA. In this section, we evaluate SMA and its variants using a set of benchmark functions and compare them with other swarm intelligent algorithms, including Whale Optimization Algorithm (WOA), Sparrow Search Algorithm (SSA), and Tuna Swarm Optimization Algorithm (TSO).

3.4.1 Experimental Parameters Settings

All algorithms are evaluated under identical conditions in comparative experiments. The population size is set to 30, and the maximum number of iterations is set to 500 for each algorithm [23, 28]. Table 1 presents the remaining common parameters for all algorithms.

Table 1. Parameters setting for each algorithm

Algorithm	Parameter settings
SMA	$z = 0.03$
GSMA	$z = 0.03, p \ge 2d + 3,$
GODSMA	$z = 0.03, R = rand, P_c = 0.5, p \ge 2d + 3$
ISMA	$z = 0.03, R = rand, P_c = 0.5, p \ge 2d + 3, t(iter)$
WOA	$a_1 = [2, 0], a_1 = [-1, -2], b = 1$
SSA	$c_1 = rand, c_2 = rand$
TSO	$z = 0.05, a = 0.7$

3.4.2 Benchmark Functions

This study utilizes a set of 12 benchmark functions to evaluate the performance of the algorithms. Functions F1 to F5, as presented in Table 2, are unimodal functions used to assess the convergence speed of the algorithms, while functions F6 to F12, as presented in Table 3, are multimodal functions employed to evaluate the algorithms' global exploration ability and ability to avoid local optima.

Table 2. Description of benchmark functions F1~F5

Test function	Optimum
$F_1(x) = \sum_{i=1}^{D} x_i^2$	0
$F_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	0
$F_3(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	0
$F_4 = \max\{\lvert x_i \rvert, 1 \le i \le D\}$	0
$F_5(x) = \sum_{i=1}^{D} i x_i^4 + random[0,1)$	0

Table 3. Description of benchmark functions F6~F12

Test function	Optimum
$F_6(x)=\sum_{i=1}^{D}-x_i\sin(\sqrt{\lvert x_i\rvert})$	-418.9829*D
$F_7(x)=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i)+10D)$	0
$F_8(x)=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)$ $-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+20+e$	8.8818e-16
$F_9(x)=\frac{1}{4000}\sum_{i=1}^{D}(x_i^2)-\left(\prod_{i=1}^{D}\cos(x_i/\sqrt{i})\right)+1$	0
$F_{10}(x)=\left(\left(\frac{1}{500}\right)+\sum_{j=1}^{25}\left(1/\left(\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6\right)\right)\right)^{-1}$	0.998004
$F_{11}(x)=\left[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)\right]$ $\times\left[30+\left(2x_1-3x_2\right)^2\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right]$	3
$F_{12}(x)=-\sum_{i=1}^{5}\left[\left(X-a_i\right)\left(X-a_i\right)^T+c_i\right]^{-1}$	-10.1532

3.4.3 Experimental Results and Analyses

The experiments were conducted on a Windows 10 64-bit home edition operating system with an AMD R7 5800H processor and 16GB RAM. The algorithms were implemented using MATLAB R2020a.

To mitigate the impact of random factors, each comparative algorithm is independently executed 30 times for each benchmark function. The optimal and worst value are recorded, and the average value (AVG) and standard deviation (STD) are calculated. The experimental results for SMA and its variants are shown in Table 4, while the experimental results for ISMA and other swarm intelligent algorithms are displayed in Table 5.

According to Table 4(a) and Table 4(b), ISMA achieves the top or joint-top ranking due to its smallest average or standard deviation when solving F1 to F5, highlighting its superior optimization capability compared to SMA, GSMA and GODSMA.

According to Table 5(a) and Table 5(b), when compared with other swarm intelligent algorithms such as WOA, TSO and SSA, ISMA consistently demonstrates the best optimization performance and minimum standard deviation when solving F1 to F5. For F6, all algorithms except WOA and SSA can achieve good optimization results. Similarly, for F7 to F11, all algorithms reach the optimal value, while for F12, all algorithms except WOA can find the optimal value. The experimental results indicate a significant improvement in optimization accuracy and stability of ISMA, clearly superior to other algorithms.

Table 4 (a). Comparison of convergence speed on F1~F5 between SMA and its variants

Function	Algorithm	Optimum	AVG	STD
F1	SMA	0	3.0515e-245	0
	GSMA	0	8.7499e-301	0
	GODSMA	0	0	0
	ISMA	0	0	0
F2	SMA	5.1441e-283	2.4259e-139	1.3287e-138
	GSMA	8.7590e-270	8.3944e-148	4.5978e-147
	GODSMA	0	0	0
	ISMA	0	0	0
F3	SMA	0	1.8119e-301	0
	GSMA	0	7.7356e-278	0
	GODSMA	0	0	0
	ISMA	0	0	0
F4	SMA	1.4068e-257	3.6686e-134	2.0094e-133
	GSMA	4.4093e-285	7.9851e-156	4.3699e-155
	GODSMA	0	0	0
	ISMA	0	0	0
F5	SMA	2.3203e-05	2.5475e-04	2.2422e-04
	GSMA	8.9907e-06	2.0529e-04	1.7774e-04
	GODSMA	8.5239e-06	3.8654e-05	4.0257e-05
	ISMA	8.2581e-07	3.2941e-05	2.7859e-05

Table 4 (b). Comparison of search ability on F6~F12 between SMA and its variants

Function	Algorithm	Optimum	AVG	STD
F6	SMA	-1.2569e+04	-1.2569e+04	0.4389
	GSMA	-1.2569e+04	-1.2568e+04	0.8004
	GODSMA	-1.2569e+04	-1.2569e+04	0.3673
	ISMA	-1.2569e+04	-1.2569e+04	0.2230
F7	SMA	0	0	0
	GSMA	0	0	0
	GODSMA	0	0	0
	ISMA	0	0	0
F8	SMA	8.8818e-16	8.8818e-16	0
	GSMA	8.8818e-16	8.8818e-16	0
	GODSMA	8.8818e-16	8.8818e-16	0
	ISMA	8.8818e-16	8.8818e-16	0
F9	SMA	0	0	0
	GSMA	0	0	0
	GODSMA	0	0	0
	ISMA	0	0	0
F10	SMA	0.9980	0.9980	2.7282e-12
	GSMA	0.9980	0.9980	3.9953e-12
	GODSMA	0.9980	0.9980	3.4045e-13
	ISMA	0.9980	0.9980	2.4251e-13
F11	SMA	3.0000	3.0000	3.3923e-10
	GSMA	3.0000	3.0000	1.9875e-10
	GODSMA	3.0000	3.0000	1.8256e-10
	ISMA	3.0000	3.0000	1.4312e-10
F12	SMA	-10.1532	-10.1527	3.2570e-04
	GSMA	-10.1532	-10.1528	6.8627e-07
	GODSMA	-10.1532	-10.1532	5.4386e-14
	ISMA	-10.1532	-10.1532	4.5348e-15

Table 5 (a). Comparison of convergence speed on F1~F5 between SMA and other swarm intelligent algorithms

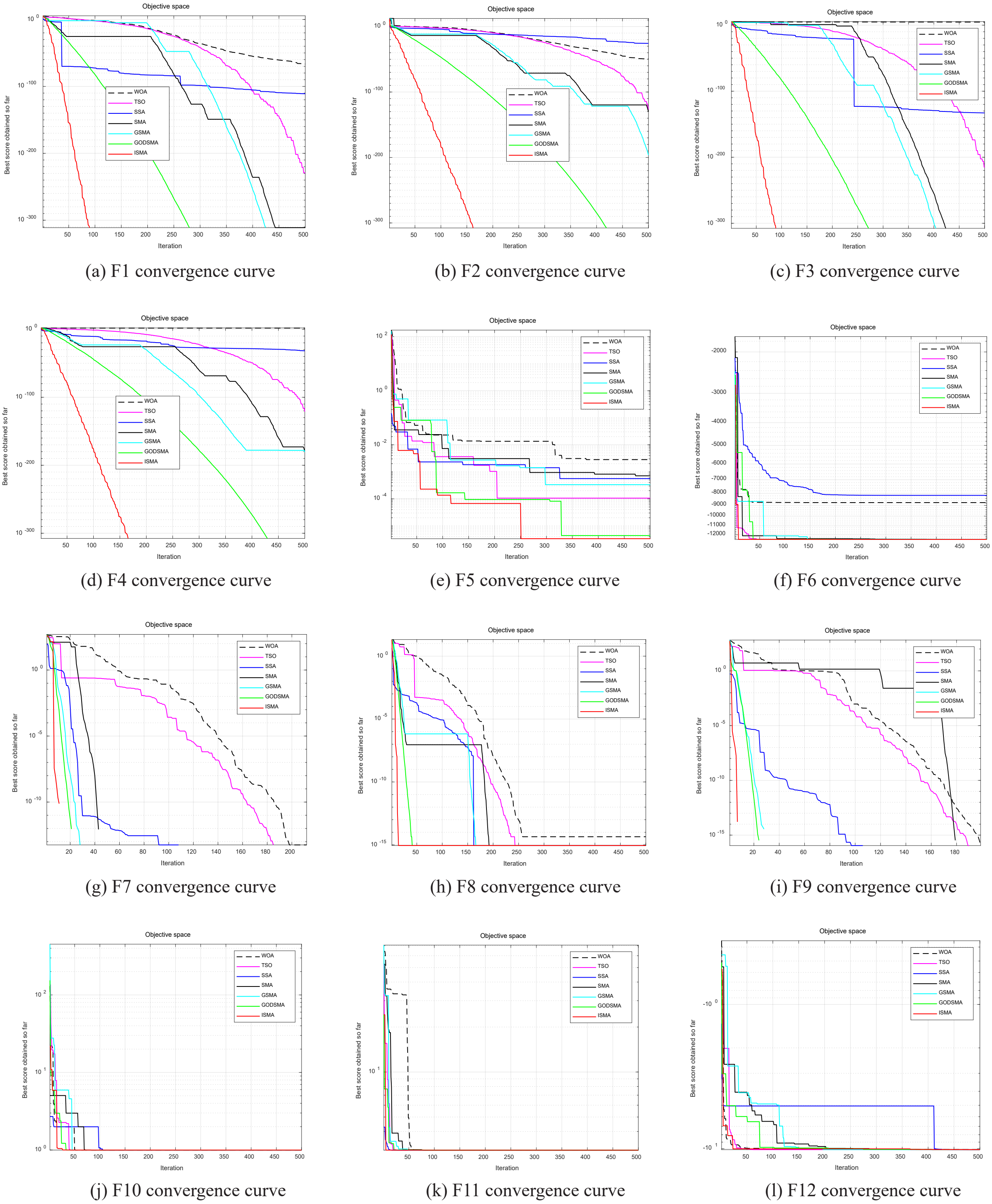
Function	Algorithm	Optimum	AVG	STD
F1	WOA	1.3392e-79	1.0730e-62	4.5434e-62
	TSO	1.6348e-266	1.8067e-210	0
	SSA	2.9162e-142	1.5758e-38	8.6167e-38
	ISMA	0	0	0
F2	WOA	4.9391e-55	8.8168e-48	2.784e-47
	TSO	1.8672e-127	2.6444e-112	1.4324e-111
	SSA	2.6416e-50	3.4838e-20	1.6328e-19
	ISMA	0	0	0
F3	WOA	2.7803e+04	6.2430e+04	1.7879e+04
	TSO	4.4077e-237	2.1948e-207	0
	SSA	2.4083e-88	4.5713e-23	2.5008e-22
	ISMA	0	0	0
F4	WOA	0.3529	49.2702	30.6570
	TSO	3.6184e-132	4.7581e-110	1.7109e-109
	SSA	0	5.4626e-18	2.1598e-17
	ISMA	0	0	0
F5	WOA	2.9674e-04	0.0063	0.0061
	TSO	7.7771e-06	4.2135e-04	3.9185e-04
	SSA	8.2321e-05	0.0014	0.0015
	ISMA	8.2581e-07	3.2941e-05	2.7859e-05

Table 5 (b). Comparison of search ability on F6~F12 between SMA and other swarm intelligent algorithms

Function	Algorithm	Optimum	AVG	STD
F6	WOA	-1.2534e+04	-1.0594e+04	2.1004e+03
	TSO	-1.2569e+04	-1.2569e+04	0.0246
	SSA	-8.9356e+03	-8.0033e+03	638.3644
	ISMA	-1.2569e+04	-1.2569e+04	0.2230
F7	WOA	0	1.8948e-15	1.0378e-14
	TSO	0	0	0
	SSA	0	0	0
	ISMA	0	0	0
F8	WOA	8.8818e-16	4.6777e-15	2.7886e-15
	TSO	8.8818e-16	8.8818e-16	0
	SSA	8.8818e-16	1.0066e-15	6.4863e-16
	ISMA	8.8818e-16	8.8818e-16	0
F9	WOA	0	0	0
	TSO	0	0	0
	SSA	0	0	0
	ISMA	0	0	0
F10	WOA	0.9980	4.0351	3.9984
	TSO	0.9980	0.9980	2.3142e-16
	SSA	0.9980	8.3360	5.3226
	ISMA	0.9980	0.9980	2.4251e-13
F11	WOA	3.0000	3.0003	5.5023e-04
	TSO	3.0000	3.0000	4.2243e-15
	SSA	3.0000	5.7000	8.2385
	ISMA	3.0000	3.0000	1.4312e-10
F12	WOA	-10.1513	-7.8576	2.9050
	TSO	-10.1532	-10.1532	4.7114e-15
	SSA	-10.1532	-8.2839	2.4987
	ISMA	-10.1532	-10.1532	4.5348e-15

Additionally, to provide a more intuitive representation of the optimization speed and accuracy of ISMA algorithm, we plot the convergence curves of the seven algorithms for F1 to F12, as depicted in Figure 1. The convergence curves for F1 to F5 demonstrate a significant improvement in the convergence speed of ISMA. Particularly for F5, ISMA reaches the optimal value within 250 iterations, outperforming other algorithms in terms of speed. Analyzing the convergence curves for F6 to F12, it is evident that ISMA discovers the global optimal value within 50 iterations, showcasing its superior global exploration ability compared to other algorithms. Furthermore, the convergence curve for F12 illustrated that ISMA identifies the optimal value faster, despite a relatively small improvement in optimization accuracy and the presence of multiple inflection points on the curve. This indicates that ISMA is easier to escape local optima value and possesses better search capability for the global optimum.

In conclusion, based on the convergence curve, the improved SMA, known as ISMA, demonstrates superior convergence speed and optimization accuracy compared to other algorithms examined in the above experiments.



(a) F1 convergence curve

(b) F2 convergence curve

(c) F3 convergence curve

(d) F4 convergence curve

(e) F5 convergence curve

(f) F6 convergence curve

(g) F7 convergence curve

(h) F8 convergence curve

(i) F9 convergence curve

(j) F10 convergence curve

(k) F11 convergence curve

(l) F12 convergence curve

Figure 1. Convergence curve of F1~F12

4 Proposed Short-Term Traffic Flow Prediction Model

In this section, we utilize ISMA to optimize the parameters of KELM and predict the short-term traffic flow on given data set, referred to as ISMA-KELM.

4.1 Modelling of ISMA-KELM

Previous studies extensively employed ELM for short-term traffic flow prediction. Here, we introduce the Gaussian radial basis function to ELM.

The prediction accuracy of the KELM model is influenced by the regularization coefficient and the kernel parameter σ. Therefore, ISMA is applied to optimize these two super parameters and improve the prediction accuracy in this study. The process of ISMA-KELM prediction model is described as follows:

Step1: Data preprocessing involves supplementing missing data and applying the moving average method to eliminate data noise. The source data is divided into training and test sets, followed by normalization.

Step2: Initialization of relevant parameters, including the population size, the maximum iteration value, dimension, and slime mould locations initialized by the Good Point Set.

Step3: Construction of the ISMA-KELM coupling model involves calculating the individual fitness $f(x_i)$, $i = 1, 2, ..., n$ of the slime mould. Subsequently, the slime mould is ranked based on its fitness value to determine the current optimal individual. The fitness function is defined as follows:

$$f(x_i) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \qquad \textbf{(14)}$$

where y_i represents the true value and $\hat{y}_i$ represents the predicted value.

Step4: Optimization of the initial slime mold individual is performed using reverse differential evolution. The selected slime mould from step 3 undergoes further optimization. The weight of each slime mould is calculated, and the current slime mould location and individual fitness value are obtained.

Step5: Updating the global optimal fitness value and optimal position involves using adaptive t-distribution to update the slime mould's position.

Step6: Outputting results occurs when the maximum number of iterations is reached, indicating completion of the optimization process. The spatial location of the optimal slime mould fitness is outputted. Otherwise, return to Step 3.

Step7: Prediction is performed on the test set using the best parameters optimized by ISMA on training set. ISMA-KELM is employed to predict the short-term traffic flow on test set.

4.2 Explanation of Data Set

Initially, we conducted experiments on the traffic flow near Heathrow Airport of M25 Expressway in the UK at 4926K observation point. The sampling period spanned from August 1, 2019 to August 25, 2019, with a 15-minute interval. The dataset comprised a total of 2400 groups of traffic flow data. The training set included data from August 1 to August 20, while the test set encompassed data from August 21 to August 25.

Subsequently, we modified the experimental dataset to assess the robustness and generalization of the model. After conducting a comparative analysis, we selected the traffic flow dataset from the 4926K observation point during a different time period. The sampling period ranged from September 1, 2019 to September 25, 2019, with a 15-minute interval. The dataset consisted of 2400 groups of traffic flow data. The training set comprised data from September 1 to September 20, while the test set consisted of data from September 21 to September 25.

4.3 Evaluation of ISMA-KELM Model

To assess the predictive performance of the ISMA-KELM model, three indices, including the Mean Absolute Error (MAE), the Mean Absolute Percentage Error (MAPE) and the Root Mean Square Error (RMSE) [29-30], are utilized to evaluate its prediction accuracy on the given traffic flow data.

4.4 Experimental Results and Analysis

To enhance the validation of prediction accuracy, we compare the ISMA-KELM model with other models including RF, LSSVM, TSO-KELM, KELM, and SMA-KELM using the provided traffic flow data. The experimental results are presented in Figure 2.

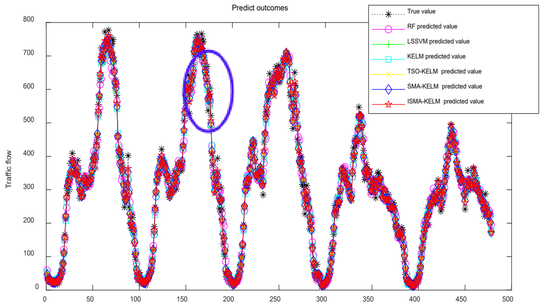


Figure 2. Prediction on 500 input samples of different models from August 1 to August 25

All six models have input vector dimension of 10 time steps (equivalent to approximately 150 minutes), and the output vector dimension is 1 time step. Consequently, the change in traffic flow for the next 15 minutes can be predicted using the given historical traffic flow data spanning two and a half hours. To identify the experimental results more clearly, we present the experimental results of different models for input samples No.160 to No.180 in Figure 3. The evaluation of these six prediction models is summarized in Table 6.

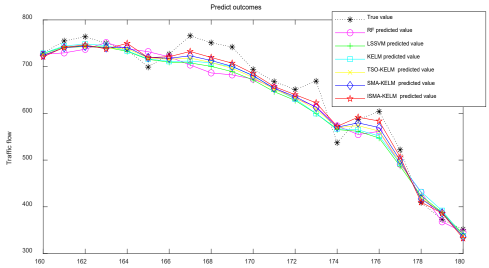


Figure 3. Prediction on No.160 to No.180 input samples of different models from August 1 to August 25

Table 6. Comparison of prediction accuracy of different models from August 1 to August 25

MODELS	MAE	MAPE/%	RMSE
RF	19.1084	9.2382	26.1524
LSSVM	15.6351	8.7914	20.9843
KELM	15.3991	8.6234	20.5391
TSO-KELM	14.1175	7.8961	18.6460
SMA-KELM	13.7498	7.6341	18.1150
ISMA-KELM	11.9747	6.6221	15.6103

Observing the curves in Figure 2 and Figure 3, it is evident that the change trend of the ISMA-KELM curve aligns more closely with the true value. Table 6 provides data illustrating that RF yields the poorest prediction results, while LSSVM and KELM exhibit similar prediction values. Both TSO-KELM and SMA-KELM show improved prediction accuracy compared to KELM. However, the proposed ISMA-KELM model outperforms KELM and SMA-KELM, demonstrating a reduction in the MAE index by 22.23% and 12.91% respectively, MAPE index by 23.21% and 15.28% respectively, and RMSE index by 24.00% and 16.05% respectively. These results indicate that the ISMA-KELM model achieves superior prediction accuracy and better alignment with real values on the provided traffic flow dataset.

Then, we repeated the same experimental process on the traffic flow data set of 4926K observation point from September 1, 2019 to September 25. We aim to verify the generalization and robustness properties of the ISMA-KEML model. The results are shown in Figure 4, Figure 5 and Table 7.

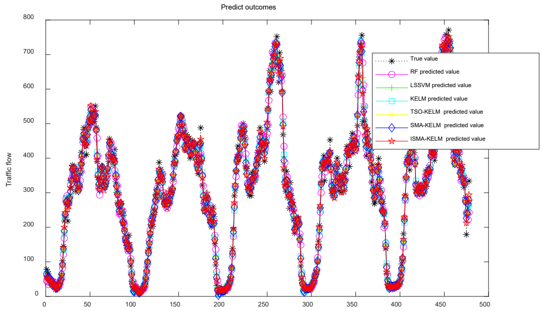


Figure 4. Prediction on 500 input samples of different models from September 1 to September 25

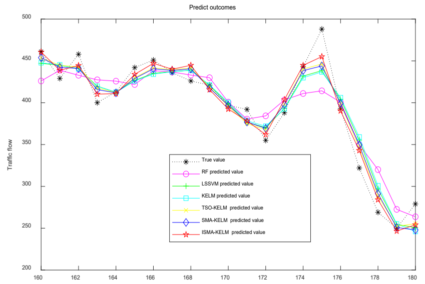


Figure 5. Prediction on No.160 to No.180 input samples of different models from September 1 to September 25

Table 7. Comparison of prediction accuracy of different models from September 1 to September 25

Models	MAE	MAPE/%	RMSE
RF	19.4605	9.9186	26.4811
LSSVM	15.0121	8.3887	20.2214
KELM	15.2359	8.7153	20.3260
TSO-KELM	13.3307	7.5958	17.9873
SMA-KELM	13.1468	7.2803	17.5185
ISMA-KELM	11.2155	6.0517	14.9753

5 Conclusion

In this paper, we have proposed a short-term traffic flow prediction model based on ISMA-KELM. Firstly, the standard slime mould algorithm is enhanced using three strategies, including good point set, reverse differential evolution and adaptive t-distribution mutation. After comparing the results of these improvements on twelve test functions, we select the improved ISAM algorithm that incorporates the aforementioned three strategies. Furthermore, the ISMA is employed to optimize the regularization coefficient and kernel function parameters in KELM, thereby enhancing prediction performance. Traffic flow datasets from observations near Heathrow Airport on the M25 Expressway in the UK is used to assess the efficiency of our proposed model in comparison to other five models. Finally, the experimental results demonstrate that ISMA-KELM exhibits a smaller prediction error on provided dataset, highlighting the superior performance and increased accuracy of the proposed model.

However, our experiment has limitations in terms of verifying the generalization and robustness of the model. Subsequent research should involve applying the proposed ISMA-KELM model to predict traffic flow and velocity in diverse datasets, as well as investing its applicability to domestic traffic flow data set.

References

[1] H.-F. Zheng, F. Lin, X.-X. Feng, Y.-J Chen, A Hybrid Deep Learning Model With Attention-Based Conv-LSTM Networks for Short-Term Traffic Flow Prediction, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 11, pp. 6910-6920, November, 2021.

[2] B. Lei, L. Wen, H. Geng, J.-M. Li, Research on Short-Term Traffic Flow Prediction based on Weighted Combinational Models, *Measurement & Control Technology*, Vol. 37, No. 5, pp. 37-41, May, 2018.

[3] W.-H. Bai, C.-B. Zhang, S.-Y. Zhang, T. Zhou, Outlier-identified Kalman Filter for Short-term Traffic Flow Forecasting, *Application Research of Computers*, Vol. 38, No. 3, pp. 817-821, March, 2021.

[4] H. Zhang, Road Network Traffic Simulation based on Short-term Traffic Flow Prediction, *Journal of Suzhou Vocational University*, Vol.32, No.1, pp. 6-10, March, 2021.

[5] H. Zhao, D.-M. Zhai, C.-H. Shi, Review of Short-term Traffic Flow Forecasting Models, *Urban Rapid Rail Transit,* Vol. 32, No. 4, pp. 50-54, August, 2019.

[6] Q. Chen, W. Wang, X. Huang, H.-N. Liang, Attention-based Recurrent Neural Network for Traffic Flow Prediction, *Journal of Internet Technology*, Vol. 21, No. 3 , pp. 831-839, May, 2020.

[7] X.-M. Ji, J.-W. Gao, X. Liu, B. Zhang, Short-Term Traffic Flow Forecasting based on Extreme Learning Machine, *Journal of Qingdao University (E&T)*, Vol.

30, No. 4, pp. 58-61, 67, December, 2015.

[8] T. Sun, H.-X. Pan, Traffic Flow Forecast based on DE and ELM, *Science & Technology Vision*, No. 30, pp. 114-115, October, 2013.

[9] T. Wang, S.-H. Xie, W.-H. Li, W.-Y. Li, Short-Term Traffic Flow Adaptive Prediction Model based on FFOS-ELM and PF, *Journal of Chongqing Jiaotong University (Natural Science)*, Vol. 40, No. 6, pp. 21-27, June, 2021.

[10] X.-P. Cui, H.-Q. Huang, Ship Traffic Flow Prediction Model based on GA-ELM Algorithm, *Microcomputer & Its Applications*, Vol. 36, No. 9, pp. 15-17, 21, May, 2017.

[11] R.-Q. Chen, J.-C. Li, J.-S. Yu, Short-term Traffic Flow Forecasting based on Hybrid FWADE-ELM, *Control and Decision*, Vol. 36, No. 4, pp. 925-932, April, 2021.

[12] B.-R. Zhou, W.-G. Cheng, Z.-Y. Xu, X.-L. Wen, A Short-term Traffic Flow Prediction Model based on Improved Online Extreme Learning Machine, *Computer Engineering & Science*, Vol. 44, No. 5, pp. 944-950, May, 2022.

[13] C. Xuan, Short-term Wind Power Prediction based on Optimized Kernel Extreme Learning Machine, *Software Engineering*, Vol. 25, No. 4, pp. 39-47, April, 2022.

[14] C.-S. Zhu, K.-P. Zhao, Wind Power Prediction of Kernel Extreme Learning Machine based on Improved Grey Wolf Algorithm, *Computer Applications and Software*, Vol. 39, No. 5, pp. 291-298, May, 2022.

[15] F.-Y. Ma, X.-X. Li, Landslide Displacement Prediction Model Using Improved SSA-KELM Coupling Algorithm, *Science Technology and Engineering*, Vol. 22, No. 5, pp. 1786-1793, February, 2022.

[16] H.-R. Chen, *Research and Application of Method for Real-time Traffic Status Identification of Highway Basic Sections*, M. D. Thesis, Chang'an University, Xi'an, China, 2015.

[17] C.-L. He, L.-P. Qu, J. Zhang, T.-L. Gao, Recognition Method of Electric Energy Disturbance based on Nuclear Extreme Learning Machine of Bee Colony Optimization, *Journal of Beihua university (Natural Science)*, Vol. 23, No. 2, pp. 265-273, March, 2022.

[18] Z.-B. Chen, Combined Prediction Model of Dynamic Online Kernel Extreme Learning Machine With Time Weight, *Statistics & Decision*, Vol. 37, No. 13, pp. 37-41, July, 2021.

[19] M. Dong, J.-S Hu, J. Yang, D.-R. Song, J.-H. Wan, An Improved Slime Mould Algorithm based MPPT Strategy for Multi-peak Photovoltaic System, *Control Theory & Applications*, Vol. 40, No. 8, pp. 1440-1448, August, 2023.

[20] R.-Q. Ren, Short-term Wind Power Forcasting based on Improved BBO Algorithm Optimizing KELM, *Wirless Internet Technology*, Vol. 17, No. 18, pp. 137-141, September, 2020.

[21] Y.-M. Xing, X.-J. Ban, R.-Y. Liu, A Short-term Traffic Flow Prediction Method based on Kernel Extreme Learning Machine, *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Shanghai, China, 2018, pp. 533-536.

[22] G.-B. Huang, H.-M. Zhou, X.-J. Ding, 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, April, 2012.

[23] S.-M. Li, H.-L. Chen, M.-J. Wang, A.-A Heidari, S. Mirjalili, Slime Mould Algorithm: A New Method for Stochastic Optimization, *Future Generation Computer Systems*, Vol. 111, pp. 300-323, October, 2020.

[24] C.-C. Gao, X.-C. Chen, R. Zhang, Q.-Y. Song, D. Yi, Y.-Z. Wu, Application of Three New Intelligent Algorithms in Epidemic Early Warning Model-COVID-19 Epidemic Warning based on Baidu Search Index, *Computer Engineering and Applications*, Vol. 57, No. 8, pp. 256-263, February, 2021.

[25] J.-H. Tian, C.-X. Li, A. Li, Medium- and Long-term Runoff Prediction based on SMA-LSSVM, *Pearl River*, Vol. 43, No. 6, pp. 101-107, June, 2022.

[26] C.-T. Zang, R.-R. Liu, H.-B. Yan, Bearing Remaining Useful Life Prediction Method based on SMA-LSTM, *Journal of Jiangsu University of Technology*, Vol. 28, No. 2, pp. 111-120, April, 2022.

[27] Y.-N. Xiao, X. Sun, S.-P. Li, J.-Y. Yao, Speed Control of Brushless Direct Current Motor based on Chaotic Elite Slime Mould Algorithm, *Science Technology and Engineering*, Vol. 21, No. 28, pp. 12130-12138, October, 2021.

[28] X. Wu, H.-X. Jiang, Y. Wu, X.-Y. Wu, J.-Y. Jiang, L. Tong, Mayfly Optimization Algorithm based on Good-point Set and Lévy Flight Principle, *Journal of Science of Teachers' College and University*, Vol. 42, No. 3, pp. 36-41, 51, March, 2022.

[29] Y.-X. Guo, S. Liu, L. Zhang, Q. Huang, Elite Opposition-based Learning Quadratic Interpolation Slime Mould Algorithm, *Application Research of Computers*, Vol. 38, No. 12, pp. 3651-3656, December, 2021.

[30] X.-P. Wang, S.-C. Chu, V. Snášel, H.-A. Shehadeh, J.-S. Pan, Five Phases Algorithm: A Novel Meta-heuristic Algorithm and Its Application on Economic Load Dispatch Problem, *Journal of Internet Technology*, Vol. 24, No. 4 , pp. 837-848, July, 2023.

Biographies

Ming Zhao received her M.S. degree in computer science from China University of Geosciences in 2004. She is currently an Associate Professor of Shanghai University of Engineering Science. Her research interests include pattern recognition, intelligent system and data mining.

Jian Tang received his M.S. degree in transportation from Shanghai University of Engineering Science in 2023. His research interests are mainly in intelligent traffic system and data mining.

Guo-Qing Chang is currently pursuing his M.S. degree in transportation from Shanghai University of Engineering Science. His research interests are mainly in intelligent traffic system and data mining.