




Article

Hybrid Extreme Learning for Reliable Short-Term Traffic Flow Forecasting

Huayuan Chen ¹, Zhizhe Lin ² , Yamin Yao ³, Hai Xie ², Youyi Song ^{1,*}  and Teng Zhou ^{1,2,4} 

¹ Faculty of Health and Social Sciences, The Hong Kong Polytechnic University, Hong Kong SAR, China; 20106306r@connect.polyu.hk (H.C.); zhouteng@csj.uestc.edu.cn (T.Z.)

² School of Cyberspace Security, Hainan University, Haikou 570228, China; linzhizhe@hainanu.edu.cn (Z.L.); xiehai@hainanu.edu.cn (H.X.)

³ Department of Computer Science, Shantou University, Shantou 515063, China; 18ymyao@alumni.stu.edu.cn

⁴ Yangtze Delta Region Institute, University of Electronic Science and Technology of China, Quzhou 324003, China

* Correspondence: youyisong@connect.polyu.hk

Abstract: Reliable forecasting of short-term traffic flow is an essential component of modern intelligent transport systems. However, existing methods fail to deal with the non-linear nature of short-term traffic flow, often making the forecasting unreliable. Herein, we propose a reliable short-term traffic flow forecasting method, termed hybrid extreme learning, that effectively learns the non-linear representation of traffic flow, boosting forecasting reliability. This new algorithm probes the non-linear nature of short-term traffic data by exploiting the artificial bee colony that selects the best-implied layer deviation and input weight matrix to enhance the multi-structural information perception capability. It speeds up the forecasting time by calculating the output weight matrix, which guarantees the real usage of the forecasting method, boosting the time reliability. We extensively evaluate the proposed hybrid extreme learning method on well-known short-term traffic flow forecasting datasets. The experimental results show that our method outperforms existing methods by a large margin in both forecasting accuracy and time, effectively demonstrating the reliability improvement of the proposed method. This reliable method may open the avenue of deep learning techniques in short-term traffic flow forecasting in real scenarios.



Citation: Chen, H.; Lin, Z.; Yao, Y.; Xie, H.; Song, Y.; Zhou, T. Hybrid Extreme Learning for Reliable Short-Term Traffic Flow Forecasting. *Mathematics* **2024**, *12*, 3303. <https://doi.org/10.3390/math12203303>

Academic Editor: Aleksandr Rakhmangulov

Received: 11 September 2024

Revised: 10 October 2024

Accepted: 15 October 2024

Published: 21 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: hybrid extreme learning; non-linear representation; artificial bee colony; short-term traffic flow forecasting

MSC: 68Q07; 68Q32; 68W50

1. Introduction

With the continuous improvement of traffic transportation systems and people's living standards, the population and vehicles in cities are increasing, which leads to traffic congestion and parking difficulties [1]. One of the promising solutions to urban traffic congestion is to develop intelligent transportation systems. Traffic flow forecasting plays an important role in intelligent transportation systems to solve traffic congestion problems [2,3], as reliable forecasting provides a solid foundation for managers to make traffic decisions. However, due to the strong nonlinearity of short-term traffic flow, it is difficult to achieve reliable forecasting [4].

Existing models for short-term traffic flow forecasting can be roughly classified into statistical analysis model, non-linear model, and artificial intelligence model [5]. Statistical analysis models include the historical average model, time series model, and Kalman filter model [6–11]. Non-linear models include mainly wavelet analysis model [12] and chaos theory model [13], whereas artificial intelligence models are developed mainly based on

neural network and deep learning [14–16]. For example, Huang et al. [17] propose a deep belief network for enhancing the ability of future representation ability.

Lv et al. [18] develop a stacked autoencoder to enhance the reliability of the forecasting model. Jia et al. [19] combined deep belief network and long short-term memory network to predict urban traffic flow under rainfall conditions.

Recently, advanced deep learning models have also been exploited to enhance forecasting reliability [20–22], but they still fail to handle the non-linear nature of short-term traffic flow [22,23]. For example, Redhu et al. [24] developed a Bi-LSTM neural network with a modified PSO technique, Chen et al. [25] designed a graph convolution network with attention mechanisms to capture the spatial–temporal features and dynamic characteristics of nodes, and Xia et al. [26] exploited a hybrid deep learning method based on residual self-attention and bidirectional gated recurrent unit combined with a convolution-gated recurrent unit network to improve the accuracy of traffic flow prediction. Although these methods are able to capture the non-linear property of short-term traffic flow, they are slow to train and computationally complex [27,28]. To improve learning efficiency, Huang et al. [29,30] leverage the random initialization technique for the input weights, hidden layer biases, and the output weight matrix with a Moore–Penrose generalized inverse algorithm, thereby easing the training difficulty. However, the extreme learning machines may face overfitting issues, leading to unreliable forecasting [31,32].

In this paper, we propose a reliable short-term traffic flow forecasting method, termed hybrid extreme learning, to effectively learn the non-linear representation of traffic flow by exploiting an artificial bee colony. Our method not only maintains competitive prediction accuracy but also reduces network complexity and avoids overfitting, yet speeds up the forecasting time, boosting the reliability. The contribution of this work is summarized as follows:

- The traffic flow forecasting model is rethought from the perspective of an artificial bee colony algorithm optimization;
- The hybrid extreme learning based on the artificial bee colony improves the forecasting reliability;
- The proposed hybrid extreme learning outperforms the state of the art, yet boosts the forecasting reliability;
- Our method shows better feasibility and superiority on the data at morning peak and afternoon low peak.

2. Methodology

In this section, we first elaborate on how to establish an extreme learning model for traffic flow forecasting. Then, we develop an artificial swarm algorithm to optimize the learning procedure.

2.1. Extreme Learning for Traffic Flow Forecasting

Given a training set $\{x_i, t_i \mid x_i \in R^D, t_i \in R^m, i = 1, 2, \dots, N\}$, where x_i represents the i th data sample, t_i represents the corresponding marker of the i th data sample, and the set represents all training data, with the assumption that the number of hidden nodes of an extreme learning machine is L , which has only one hidden layer, and whose parameters include input weight iw , output weight β , and hidden layer deviation b , we first randomly generate the input weight iw and the hidden layer deviation b . Then, we calculate $H(x)$ as the hidden layer output according to the randomly generated iw and b . In detail, $H(x)$ is calculated as follows:

$$H(x) = [h_1(x), \dots, h_L(x)]. \quad (1)$$

The output of the hidden layer is multiplied by the input and the corresponding weight, then added with the corresponding deviation, and summed by the non-linear function to get all the node results. Note that $H(x) = [h_1(x), \dots, h_L(x)]$ is the non-linear mapping of

the hidden layer output matrix of ELM as the output of the i th hidden layer node, and the output function of the hidden layer is not unique; different output functions can be used for different hidden layer neurons. In practice, $h_i(x)$ can be described as follows:

$$h(x) = g(w_i, b_i, x), \quad (2)$$

where $iw_i \in R^D$ and $b_i \in R$; $g(w_i, b_i, x)$ is essential to the extreme learning activation function [33]. The general activation function is a continuous, non-linear function commonly used in regression prediction sigmoid functions, Gaussian functions, etc. In this work, we use the sigmoid function as the activation function, i.e.,

$$g(x) = \frac{1}{1 + e^{-x}}, \quad (3)$$

where $x = iw_i x + b_i$. The output layer then is calculated as

$$f_L = \sum_{i=1}^L \beta_i h_i(x) = h(x)\beta, \quad (4)$$

where $\beta = [\beta_1, \dots, \beta_L]^T$ is the collection of the hidden layer nodes (L nodes) and the output layer (m nodes, $m \geq 1$) between the output of the weight. The operation of the neural network from the input to output is calculated by Equation (4). It should be noted, that at present, the unknowns in the above equation are iw, b, β , which are, respectively, the weights, deviations of hidden layer nodes, and output weights of hidden layer nodes [29]. Neural network learning is to adjust the weights and biases of the network in the training process, and the actual training results are contained in the connection weights and biases. In the training process, extreme learning only needs to solve the output weight β of the network, so it is a linear parameter model that can easily converge to the global optimal solution value in the training process. Assuming that the model has N input samples, the number of hidden layer nodes is L , and the number of output samples is M , the training process is as follows [29]:

- (1) Initialize the node parameters: at the beginning of training, the node parameters are determined randomly through an arbitrary continuous probability distribution, and the random generation can meet any continuous probability distribution;
- (2) Calculate the output matrix of the hidden layer: after iw and b are determined, the output H of the hidden layer can be calculated according to Equations (1) and (2). To effectively train the weights of the output β , the goal is to ensure that the training error (such as RMSE) is minimal. We here use the network output $H\beta$ and sample tag T by the least square calculation. Note that the error calculation is as $\min \|H\beta - T\|^2, \beta \in R^{L \times M}$, where H is the output of the hidden layer and T is the target matrix of the training data, namely the sample label, respectively; H can be written as follows:

$$H = [h(x_1), h(x_2), \dots, h(x_N)]^T = \begin{bmatrix} h_1(x_1) & \cdots & h_L(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_N) & \cdots & h_L(x_N) \end{bmatrix} T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}; \quad (5)$$

- (3) Solve the output weight: by deducing Equation (5), the optimal solution can be obtained by computing the following equation:

$$\hat{\beta} = H^+ T, \quad (6)$$

where H^+ is the Moore–Penrose generalized inverse of matrix H . The key to extreme learning is to find the output weight that minimizes the error to obtain the optimal solution.

2.2. Artificial Bee Colony Algorithm

It is a global optimization algorithm based on the swarm intelligence proposed by Karaboga [34,35] in 2005, inspired by the swarm foraging model proposed in Tereshko and Loengarov's article [36], which mainly simulates the collection process of bees. Recently, swarm intelligence has been widely used in complex problems. It also has developed many advanced versions, for example, the ant colony algorithm [37], which is inspired by ant foraging behavior, the particle swarm optimization algorithm inspired by the foraging behavior of birds [38], and the wolf pack algorithm inspired by the living behavior of bionic wolves [39], which was optimized into gray wolf algorithm [40,41]. In summary, the principle of the artificial bee colony algorithm is as follows (Algorithm 1):

Algorithm 1: Artificial bee colony (ABC) algorithm.

```

Data: Set ABC parameters
Data: Initialize food source location
Data: Construct the current global optimal solution BestSol
1 while  $it < MaxIt$  do
2   Leading bees procedure Following bees procedure Compare and update the
   global optimal solution BestSol Detecting bees procedure  $it = it + 1$ 
3 end
4 Procedure Employ bees:
5   for all Employed Bees do
6     Search for a new food source near the corresponding food source  $pop(i)$ 
     Calculate fitness values for new food sources  $newbee.Cost$  if the fitness
     value of the original food source  $pop(i).Cost \geq newbee.Cost$  then
7        $pop(i) = newbee$ 
8     end
9   end
10 return
11 Procedure Follow bees:
12   for all Following Bees do
13     Select food sources according to the roulette wheel selection algorithm
     Search for a new food source near the corresponding food source  $pop(i)$ 
     Calculate fitness values of new food sources for the cost of new bee
14     if the fitness value of the original food source  $pop(i).Cost \geq newbee.Cost$  then
15        $pop(i) = newbee$ 
16     end
17   end
18 return
19 Procedure Detect bees:
20   for all Employed Bees do
21     if neighborhood search times  $C(i) \geq L$  then
22       reinitialize food source  $pop(i)$   $C(i)=0$ 
23     end
24   end
25 return

```

- (1) Initialize the nectar source: first, we define $nPop$ as the number of nectar sources, and the quality of the nectar source (whether it attracts bees) is equivalent to the solution of the function, which is often referred to as the fitness value $pop.cost$ in heuristic algorithms. For this $nPop$ nectar source, we can determine its location by ourselves according to Equation (8):

$$X_{id} = L_d + rand(0, 1)(U_d - L_d); \quad (7)$$

- (2) Update the nectar source: the leader bee searches for a new nectar source around the nectar source i according to Equation (9):

$$x_{id}^{new} = x_{id} + a * \varphi * (x_{id} - x_{jd})(j \neq i), \quad (8)$$

where φ is a random number in the range $[-1, 1]$ that determines the degree of interference; a is the acceleration coefficient (usually taken as 1). If the fitness of the new nectar source is higher than that of the original nectar source, the greedy selection strategy is adopted to determine the better nectar source and replace the original nectar source with the new better nectar source;

- (3) The follower bee chooses the leader bee: according to roulette rules [42], the probability of following bees choosing leading bees is calculated, and the probability is calculated by Equation (10):

$$p_i = \frac{fit_i}{\sum_{i=1}^{nPop} fit_i}; \quad (9)$$

- (4) Produce scout bees: in the process of neighborhood search [43], if the nectar source is not updated to a better nectar source after C_i reaches the threshold L , the nectar source will be abandoned. The scout bee will randomly find a new nectar source according to Equation (11). The location of the nectar source found by the scout bee can be determined by

$$x_i = \begin{cases} L_d + rand(0, 1)(U_d - L_d) & C_i \geq L \\ x_i & C_i < L. \end{cases} \quad (10)$$

2.3. Hybrid Extreme Learning

After the ABC optimization algorithm is introduced, we use the iw and b of the extreme learning model and take the RMSE of the training set as the goal to be optimized so that the optimal parameter selection of the extreme learning becomes an optimization problem. The ABC optimization algorithm is used to obtain the optimal solution of this optimization problem, which is the best parameter combination of extreme learning model [44]. The general process of the hybrid extreme learning is as follows:

- (1) Normalize traffic flow, and construct the training and testing samples;
- (2) Construct the ABC-ELM traffic flow prediction model;
- (3) Initialize ABC parameters (population size $nPop$, maximum number of iterations $MaxIt$, and neighborhood search threshold L);
- (4) Initialize the nectar source location, and set up the nectar source location optimization range ($iw \in [-1, 1]$ and $b \in [0, 1]$);
- (5) Determine the fitness function of ABC optimization RMSE and calculate Equation (12) and its values, looking for honey's optimal location and the corresponding optimal adaptive value;
- (6) Continuously iterate and greedily select, and update the best location of nectar source;
- (7) Determine whether the RMSE reaches the minimum value or $MaxIt$. If the conditions are met, then the artificial bee colony stops iteration and outputs the best iw and b ; if not, the optimal nectar source location and its adaptive value are continued iteratively. Note that Figure 1 shows the flow chart of the proposed hybrid extreme learning method.

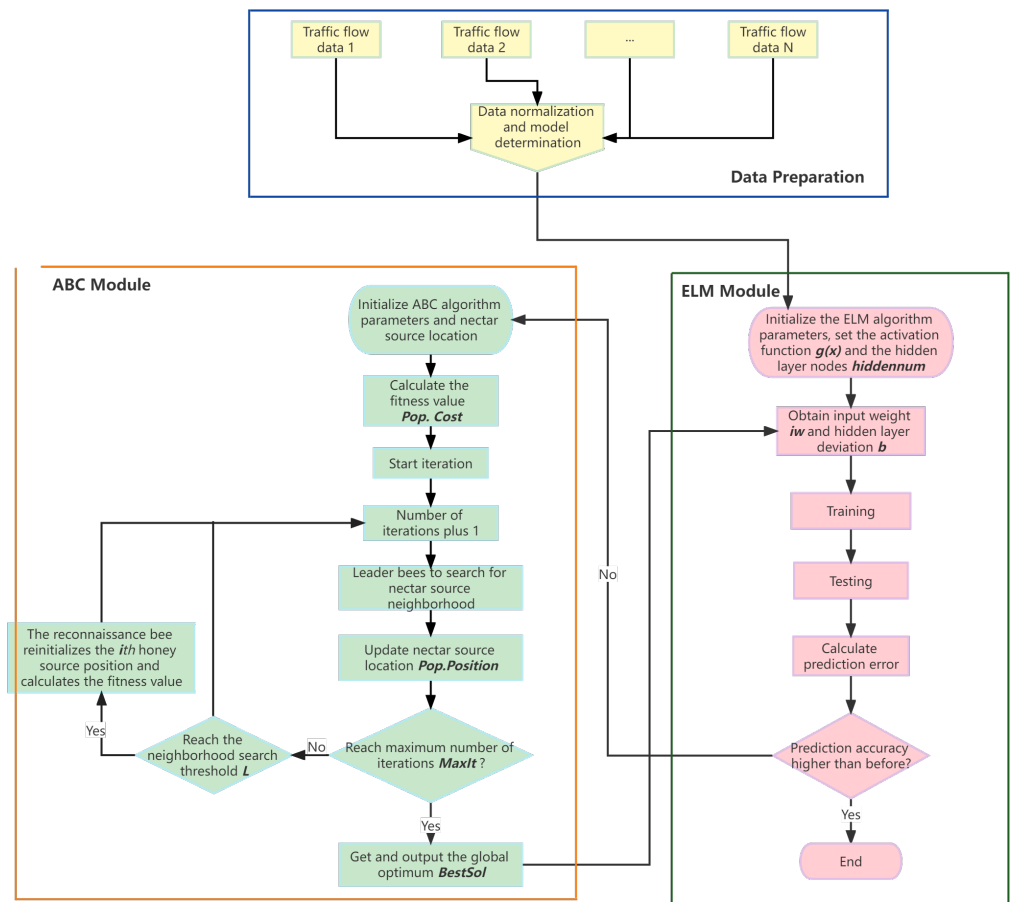


Figure 1. Flow chart of ABC-ELM model.

3. Experiments

To evaluate the performance of the proposed hybrid extreme learning for short-term traffic flow forecasting, we use the datasets collected by Wang et al. [45] from four monitoring points at the endpoints of four expressways (A1, A2, A4, A8) in Amsterdam for experiments. As shown in Figure 2, the A1, A2, A4, and A8 motorways are all connected to the A10 motorway, and all end at the A10 motorway.



Figure 2. Amsterdam's A1, A2, A4, and A8 motorways connected to the A10 ring road.

3.1. Description of Data

The data were collected using the MONICA sensor between 5 p.m. on 20 May 2010 and 5 p.m. on 24 June 2010. The data were collected over a 10-min interval, which estimates hourly traffic flow based on the number of vehicles that passed during the 10 min. In this paper, the original data are divided into two parts. The first part is the data from the first four weeks of training the model, and the second part is the data from the fifth week of testing the model. Note that the generalization error (*RMSE* and *MAPE*) after ABC-ELM model training is calculated for the assessment.

3.2. Evaluation Criteria

To evaluate the advantages and disadvantages of the short-term traffic flow prediction model, the prediction results of the model are generally compared with the actual results, and the appropriate prediction and evaluation standards are used to test. We use two criteria to evaluate the performance of the traffic flow prediction methods, e.g., root mean square error (*RMSE*) and mean absolute percent error (*MAPE*). They are defined as the following:

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (\hat{y}(n) - y(n))^2}, \quad (11)$$

$$MAPE = \frac{1}{N} \sum_{n=1}^N \left| \frac{\hat{y}(n) - y(n)}{y(n)} \right| \times 100\%, \quad (12)$$

where N represents the number of samples, $y(n)$ represents the actual traffic flow, and $\hat{y}(n)$ represents the prediction result obtained using the prediction model; *RMSE* is an evaluation criterion that can reflect not only the deviation between predictions and ground truth, but also the degree of deviation between predictions and ground truth. *MAPE* is the relative deviation between predictions and ground truth. The lower the value of *MAPE* and *RMSE*, the better the prediction.

3.3. Analysis and Discussions

3.3.1. Experimental Results

Figure 3 shows the comparison between the prediction and the ground truth of the proposed method and the corresponding prediction error. In each figure, red represents the real ground truth, blue represents the prediction of our model, and green represents the correlation error, that is, the correlation error obtained by dividing the absolute value of the difference between the prediction and the real value by the actual ground truth. It can be seen from this figure that the prediction error of the proposed model is relatively ideal in most cases. The longitudinal analysis in Figure 4 shows that the times with high correlation errors are concentrated in traffic flows with low traffic volumes but with some volatility, usually just a value jump in the forecast series (for example, Friday morning of week 5 in the figure).

During these hours, traffic flow is low, only about 1/10 of the peak hours. Thus, although the prediction residual is small, it is clearly reflected in the correlation error. Based on actual measurements, this situation also exists in other traffic flow prediction models. However, in real life, the traffic pressure of different traffic segments is lower, and the actual resources are richer, so the prediction bias will be less sensitive. In the peak period, when the prediction accuracy is high, the model has a certain superiority.

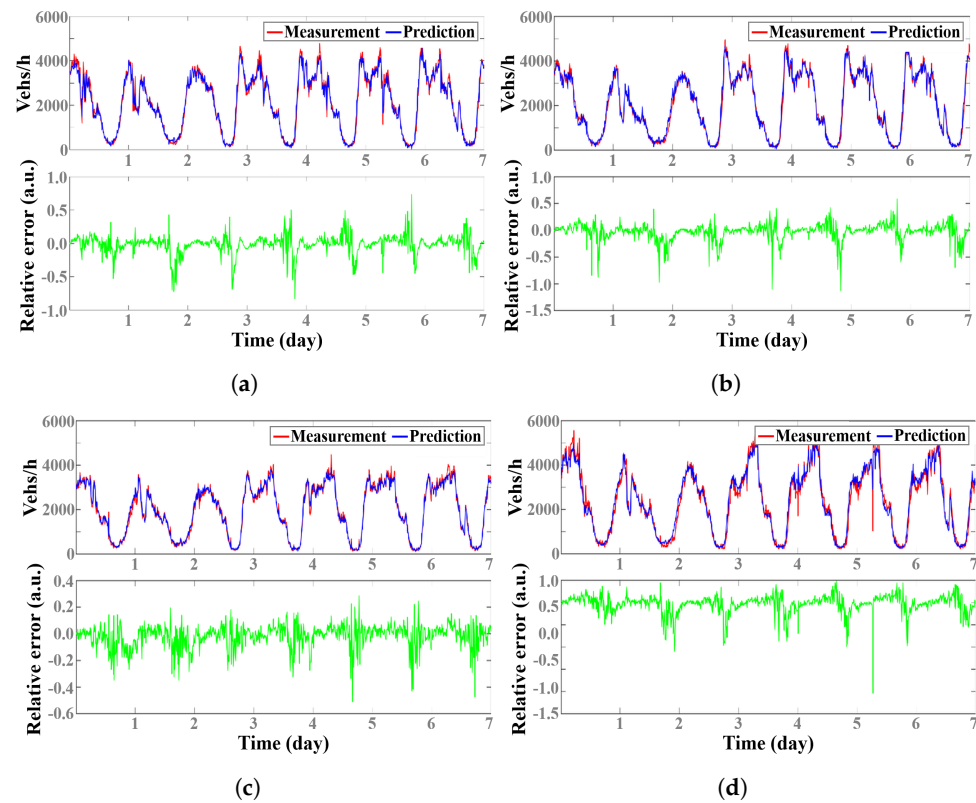


Figure 3. Figure (a–d) show the prediction results and relative errors of the ABC-ELM model on A1, A2, A4, and A8 datasets, respectively.

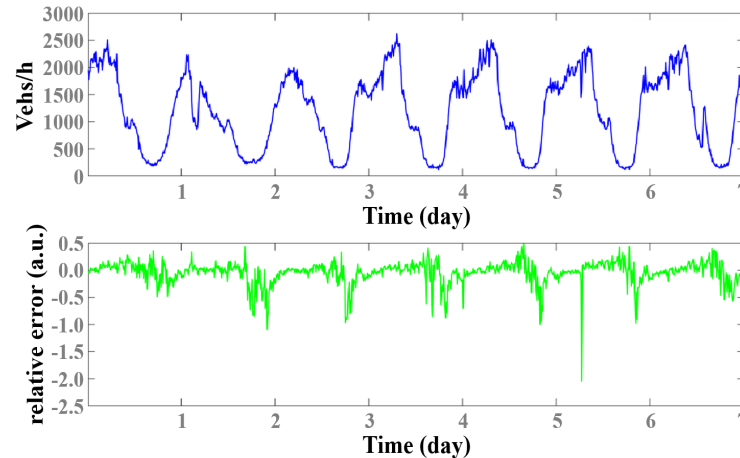


Figure 4. Ground truth and relative errors on test set A8.

3.3.2. Results Contrast

In this subsection, we use *RMSE* and *MAPE* as the measurement criteria to compare the prediction accuracy of the proposed method and commonly used short-term traffic flow prediction models in ITS on the four datasets A1, A2, A4, and A8. Smaller values of *RMSE* and *MAPE* indicate better accuracy of the prediction model. The prediction results of our model are given in Section 3.3.1. Other prediction models and relevant reference settings are as follows:

- Autoregressive model (AR) [8]: a linear regression model that describes random variables at a certain time through linear combinations of random variables within a certain time, that is, a time series analysis and prediction method. Order p set to 8.
- Seasonal autoregressive integrated moving averaging (SARIMA) [11]: a time series analysis and prediction method. The parameter is set to SARIMA (1, 0, 1) \times (0, 1, 1), 1008.

- Exponential smoothing algorithm (ES) [46]: a time series analysis and prediction method. Experiments using quadratic exponential smoothing algorithm $\alpha = 0.4$.
- Radial basis function neural network (RBFNN) [47]: a non-parametric learning model, which is composed of connected neurons.
- Kalman filter (KF) [6]: a short-term traffic flow prediction method based on Kalman filter combined with discrete wavelet analysis proposed by Xie Yuanchang [48].
- Stack autoencoder (SAE) [49]: a deep learning neural network, which was implemented with Keras module referring to Lv Yisheng's article [18] in the experiment.
- Extreme learning machine (ELM) [29]: a learning scheme of feed forward neural network was implemented with the suggestion in the original article.
- Bidirectional long short-term network (Bi-LSTM) [24]: a deep learning neural network based on optimized bidirectional long short-term network was implemented on PyTorch platform with the suggested configuration.
- Matrix-based graph neural network (MGNN) [25]: a graph neural network with matrix-based attention mechanism was implemented by the source codes provide by the authors.
- SAB-ConvGRU (RSAB) [26]: a hybrid deep learning method whose results are reproduced by implementing the codes provided by the authors with the suggested parameters.

Table 1 shows the prediction effects of different prediction models on datasets A1, A2, A4, and A8. It can be seen that for the four datasets A1, A2, A4, and A8, the prediction effect of our model is better than other prediction models. For example, compared with the relatively good performance of ES among several other models, the MAPE of our proposed model on A1, A2, A4, and A8 datasets is reduced by 1.54%, 0.18%, 1.31%, and 1.89%, respectively. Compared with RBFNN, the MAPE of our model on the A1, A2, A4 and A8 datasets is reduced by 2.21%, 0.32%, 2.13%, and 2.42%, respectively. ELM and RBFNN have similar performances due to their similar single-hidden layer network structure. Compared with RBFNN, ELM has higher learning speed, better generalization ability, and simpler structure and can be used in feedforward neural networks with more activation functions.

Table 1. Comparison of MAPE and RMSE between different prediction models.

Model	Criterion	A1	A2	A4	A8
AR [8]	RMSE (vehs/h)	301.44	214.22	226.12	166.71
	MAPE (%)	13.57	11.59	12.70	12.71
SARIMA [11]	RMSE (vehs/h)	308.44	221.08	228.36	169.36
	MAPE (%)	12.81	11.25	12.05	12.44
ES [46]	RMSE (vehs/h)	315.82	226.40	237.76	174.67
	MAPE (%)	11.94	10.75	11.97	12.00
RBFNN [47]	RMSE (vehs/h)	299.64	212.95	225.86	166.50
	MAPE (%)	12.61	10.89	12.49	12.53
KF [6]	RMSE (vehs/h)	332.03	239.87	250.51	187.48
	MAPE (%)	12.46	10.72	12.62	12.63
SAE [49]	RMSE (vehs/h)	324.52	243.99	228.47	159.65
	MAPE (%)	19.12	14.04	12.91	11.36
ELM [29]	RMSE (vehs/h)	260.16	216.34	216.18	119.40
	MAPE (%)	14.72	11.46	11.65	11.42
Bi-LSTM [24]	RMSE (vehs/h)	253.84	212.45	207.26	115.39
	MAPE (%)	12.83	10.97	11.13	10.69
MGNN [25]	RMSE (vehs/h)	247.82	213.11	205.68	113.81
	MAPE (%)	12.48	10.86	11.26	10.97
RSAB [26]	RMSE (vehs/h)	235.27	209.84	212.04	115.33
	MAPE (%)	13.65	11.21	11.47	11.37
ABC-ELM	RMSE (vehs/h)	194.19	166.78	154.23	99.31
	MAPE (%)	10.40	10.57	10.36	10.11

3.3.3. Case Study

To further make a more convincing comparison, we compared the prediction results of ABC-ELM and PSO-ELM [50], which is a hybrid prediction model of ELM optimized by particle swarm optimization (PSO), and the parameters such as population size, search

range, and the number of iterations are consistent with those of ABC-ELM. Comparison results of prediction criteria $RMSE$ and $MAPE$ are shown in Figures 5 and 6. We can see that, under the same number of iterations and other parameters, the ABC-ELM model achieves better $RMSE$ and $MAPE$ performances than the PSO-ELM model.

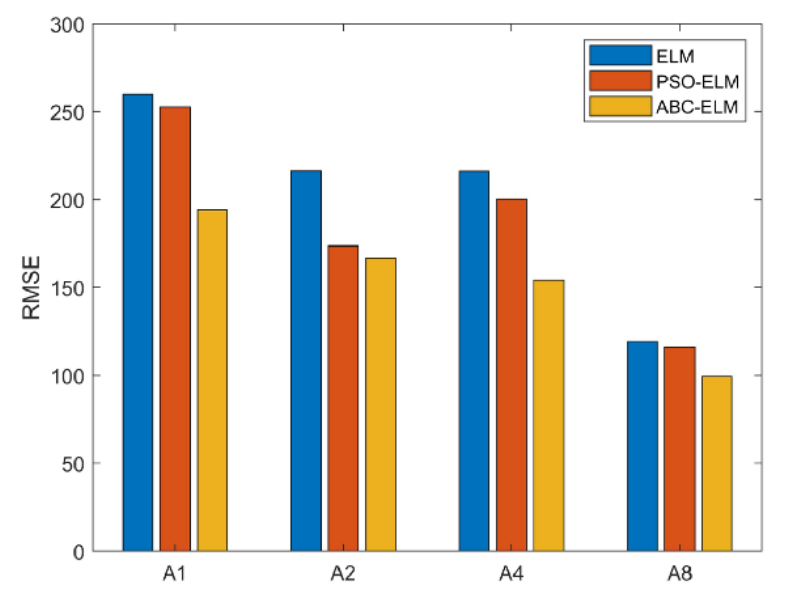


Figure 5. Comparison of $RMSE$ prediction between ABC-ELM and PSO-ELM traffic flow prediction models on datasets A1, A2, A4, and A8.

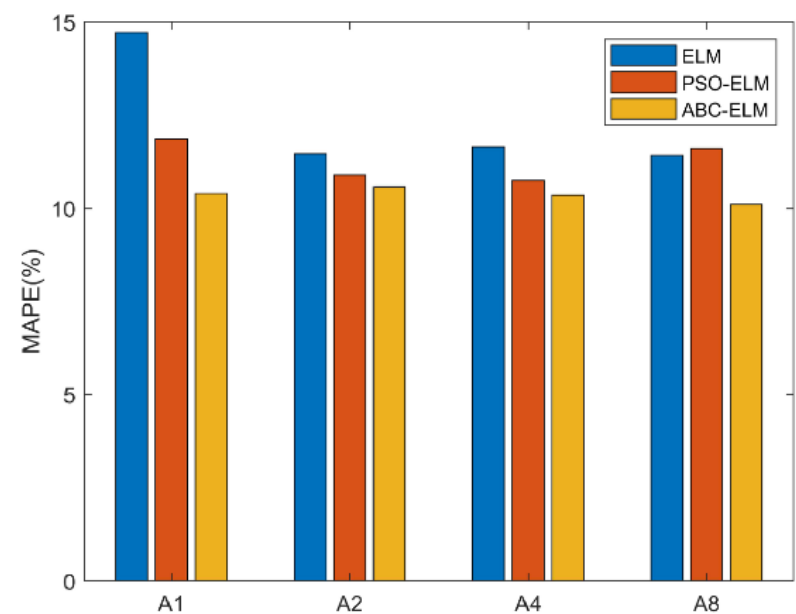


Figure 6. Comparison of $MAPE$ prediction between ABC-ELM and PSO-ELM traffic flow prediction models on datasets A1, A2, A4, and A8.

This paper proves not only that the overall optimization effect of the ABC-ELM model is better than that of other algorithmic optimization prediction models, but also that the ABC-ELM model is better than other algorithmic optimization models in different special cases. In the following, we compare the prediction results of the ABC-ELM model and PSO-ELM model in the afternoon peak period and the evening low peak period. The comparison results of the afternoon peak period are shown in Table 2 and Figure 7, and the comparison results of the evening low peak period are shown in Table 3 and Figure 8.

Table 2. Prediction performance of different optimized ELM models in the afternoon peak.

Time	Ground Truth	ELM	Prediction PSO-ELM	ABC-ELM
6/17/17:00	3561	3410	3618	3458
6/17/17:30	3546	3458	3568	3467
6/17/18:00	4014	3589	3856	3682
6/18/17:00	4027	3714	3787	3933
6/18/17:30	3948	3490	3985	3825
6/18/18:00	3370	3180	3446	3320
6/23/17:00	4337	3886	4254	4166
6/23/17:30	4884	4129	3684	4451
6/23/18:00	4182	3758	4062	4001
RMSE (vehs/h)		409.36	419.62	211.61
MAPE (%)		8.70	4.96	4.16

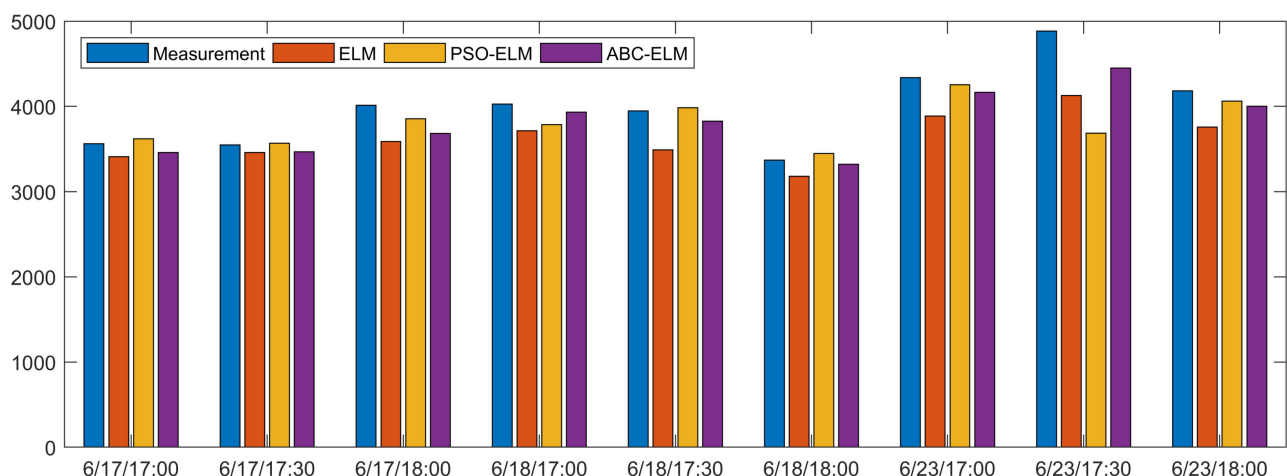


Figure 7. Prediction results of ELM optimized by different algorithms in the afternoon peak.

Table 3. Prediction performance of different optimized ELM models in low peak period at night.

Time	Ground Truth	ELM	Prediction PSO-ELM	ABC-ELM
6/20/23:00	241	287	353	286
6/20/23:30	218	342	221	252
6/21/00:00	250	301	248	277
6/21/23:00	443	403	406	449
6/21/23:30	283	519	370	304
6/22/00:00	293	520	379	311
6/22/23:00	189	212	134	203
6/22/23:00	207	227	44	221
6/23/00:00	224	289	217	255
RMSE (vehs/h)		122.05	73.04	42.92
MAPE (%)		35.23	22.25	12.86

Tables 2 and 3 list the prediction results of the ELM model optimized by different algorithms in the afternoon peak and night low peak periods. Figures 7 and 8 show the difference in the prediction results, from which we can see that ABC has better optimization ability than PSO for the ELM traffic flow prediction model. Therefore, ABC is better than PSO in optimizing ELM model parameters for traffic flow prediction under different traffic flow conditions. We further compare different algorithms that optimize ELM over more time periods to reduce the randomness of the prediction performance, as shown in Figure 9. What is clear is that the ABC-ELM studied in this paper still makes very accurate predictions when traffic flow is volatile.

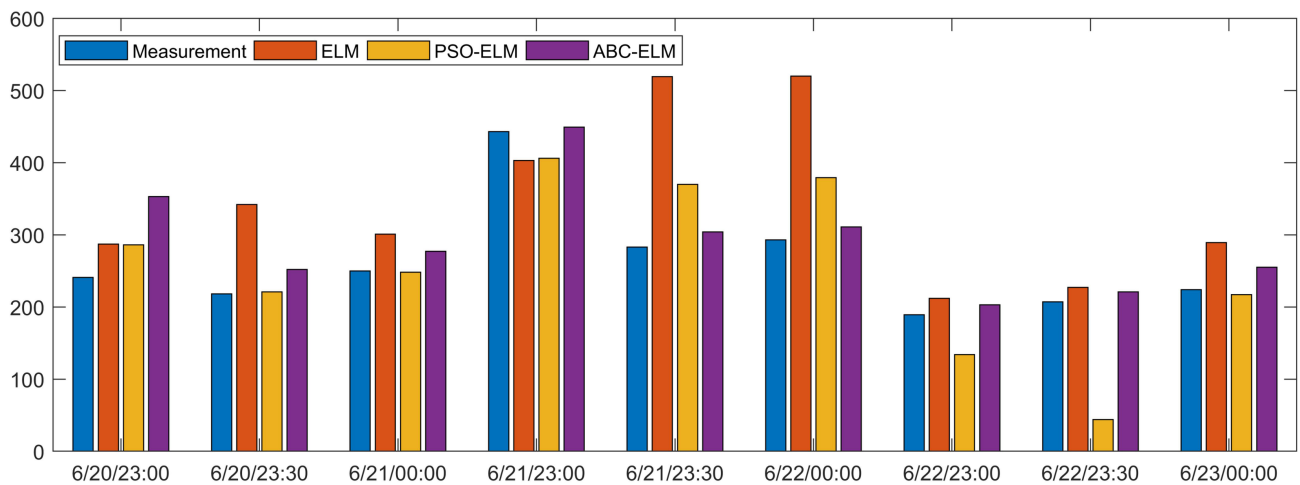


Figure 8. Prediction results of different algorithms for optimizing ELM at low peak at night.

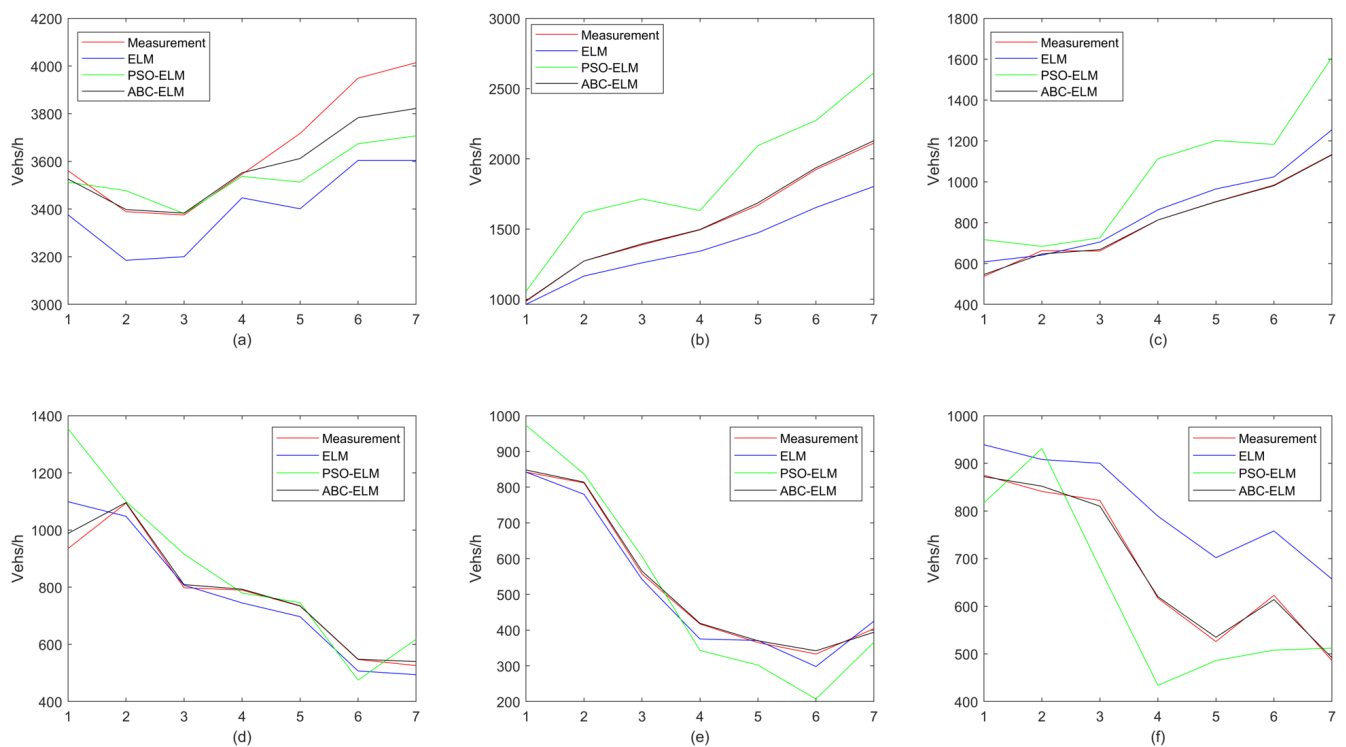


Figure 9. Figure (a–f) show the prediction results of ELM optimization by different algorithms in the case of large fluctuation of traffic flow.

4. Conclusions

In this paper, we propose an effective hybrid extreme learning method based on an artificial bee colony to select optimal input weights and hidden layer thresholds for reliable short-term traffic flow forecasting. Because of growing car ownership and urban traffic jams, the proposed method mixes traffic flow forecasting algorithms and performs a more reliable forecasting of traffic flow. Reliable traffic prediction information provides transportation for traffic management decision making with a solid foundation and guides the driver to choose a smoother line to avoid or alleviate congestion. Extensive experiments with four benchmark real-world datasets are conducted to evaluate the feasibility and superiority of the proposed forecasting model, with the positive results demonstrating its outperformance.

Author Contributions: Methodology, H.C. and Y.S.; project administration, Y.S. and T.Z.; resources, T.Z.; software, H.C. and Y.Y.; supervision, Y.S. and T.Z.; visualization, Z.L. and H.X.; writing—original draft preparation, H.C., Z.L. and Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the Natural Science Foundation of China (No. 62462021), the Natural Science Foundation of Guangdong Province (No. 2024A1515011676, No. 2022A1515011590), the Philosophy and Social Sciences Planning Project of Zhejiang Province (No. 25JCXK006YB) and the Hainan Province Higher Education Teaching Reform Project (No. HNJG2024ZD-16).

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request. Note that the source codes will be available upon publication.

Acknowledgments: The authors express their gratitude to the reviewers and editors for their valuable feedback and contributions in refining this manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Chen, Z.G.; Zhan, Z.H.; Kwong, S.; Zhang, J. Evolutionary computation for intelligent transportation in smart cities: A survey. *IEEE Comput. Intell. Mag.* **2022**, *17*, 83–102. [\[CrossRef\]](#)
- Wang, J.; Boukerche, A. Non-parametric models with optimized training strategy for vehicles traffic flow prediction. *Comput. Netw.* **2021**, *187*, 107791. [\[CrossRef\]](#)
- Ruan, H.; Wu, B.; Li, B.; Chen, Z.; Yun, W. Expressway exit station short-term traffic flow prediction with split traffic flows according originating entry stations. *IEEE Access* **2021**, *9*, 86285–86299. [\[CrossRef\]](#)
- Sun, P.; Aljeri, N.; Boukerche, A. Machine learning-based models for real-time traffic flow prediction in vehicular networks. *IEEE Netw.* **2020**, *34*, 178–185. [\[CrossRef\]](#)
- Boukerche, A.; Wang, J. A performance modeling and analysis of a novel vehicular traffic flow prediction system using a hybrid machine learning-based model. *Ad Hoc Netw.* **2020**, *106*, 102224. [\[CrossRef\]](#)
- Yin, W.; Tivay, A.; Hahn, J.O. Hemodynamic monitoring via model-based extended kalman filtering: Hemorrhage resuscitation and sedation case study. *IEEE Control Syst. Lett.* **2022**, *6*, 2455–2460. [\[CrossRef\]](#)
- Okutani, I.; Stephanedes, Y.J. Dynamic prediction of traffic volume through Kalman filtering theory. *Transp. Res. Part B Methodol.* **1984**, *18*, 1–11. [\[CrossRef\]](#)
- Domps, B.; Marmain, J.; Guérin, C.A. A Reanalysis of the October 2016 “Meteotsunami” in British Columbia with Help of High-Frequency Radars and Autoregressive Modeling. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [\[CrossRef\]](#)
- De Gooijer, J.G. Asymmetric vector moving average models: Estimation and testing. *Comput. Stat.* **2021**, *36*, 1437–1460. [\[CrossRef\]](#)
- Zhang, Y.; Yamamoto, M.; Suzuki, G.; Shioya, H. Collaborative Forecasting and Analysis of Fish Catch in Hokkaido from Multiple Scales by Using Neural Network and ARIMA Model. *IEEE Access* **2022**, *10*, 7823–7833. [\[CrossRef\]](#)
- Wang, F.; Liang, Y.; Lin, Z.; Zhou, J.; Zhou, T. SSA-ELM: A hybrid learning model for short-term traffic flow forecasting. *Mathematics* **2024**, *12*, 1895. [\[CrossRef\]](#)
- Chen, Y.C.; Li, D.C. Selection of key features for PM_{2.5} prediction using a wavelet model and RBF-LSTM. *Appl. Intell.* **2021**, *51*, 2534–2555. [\[CrossRef\]](#)
- Anter, A.M.; Gupta, D.; Castillo, O. A novel parameter estimation in dynamic model via fuzzy swarm intelligence and chaos theory for faults in wastewater treatment plant. *Soft Comput.* **2020**, *24*, 111–129. [\[CrossRef\]](#)
- Mahadeva, R.; Kumar, M.; Patole, S.P.; Manik, G. Desalination Plant Performance Prediction Model Using Grey Wolf Optimizer Based ANN Approach. *IEEE Access* **2022**, *10*, 34550–34561. [\[CrossRef\]](#)

15. Mahajan, S.; HariKrishnan, R.; Kotecha, K. Prediction of network traffic in wireless mesh networks using hybrid deep learning model. *IEEE Access* **2022**, *10*, 7003–7015. [\[CrossRef\]](#)
16. Essien, A.; Petrounias, I.; Sampaio, P.; Sampaio, S. A deep-learning model for urban traffic flow prediction with traffic events mined from twitter. *World Wide Web* **2021**, *24*, 1345–1368. [\[CrossRef\]](#)
17. Huang, W.; Song, G.; Hong, H.; Xie, K. Deep architecture for traffic flow prediction: Deep belief networks with multitask learning. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2191–2201. [\[CrossRef\]](#)
18. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F.Y. Traffic flow prediction with big data: A deep learning approach. *IEEE Trans. Intell. Transp. Syst.* **2014**, *16*, 865–873. [\[CrossRef\]](#)
19. Jia, Y.; Wu, J.; Xu, M. Traffic flow prediction with rainfall impact using a deep learning method. *J. Adv. Transp.* **2017**, *2017*. [\[CrossRef\]](#)
20. Kozhoridze, G.; Dor, E.B.; Sternberg, M. Assessing the Dynamics of Plant Species Invasion in Eastern-Mediterranean Coastal Dunes Using Cellular Automata Modeling and Satellite Time-Series Analyses. *Remote Sens.* **2022**, *14*, 1014. [\[CrossRef\]](#)
21. Wen, Y.; Xu, P.; Li, Z.; Xu, W.; Wang, X. RPConvformer: A novel Transformer-based deep neural networks for traffic flow prediction. *Expert Syst. Appl.* **2023**, *218*, 119587. [\[CrossRef\]](#)
22. Xu, Z.; Yuan, J.; Yu, L.; Wang, G.; Zhu, M. Machine Learning-Based Traffic Flow Prediction and Intelligent Traffic Management. *Int. J. Comput. Sci. Inf. Technol.* **2024**, *2*, 18–27. [\[CrossRef\]](#)
23. Zeng, W.; Wang, K.; Zhou, J.; Cheng, R. Traffic Flow Prediction based on hybrid deep learning models considering missing data and multiple factors. *Sustainability* **2023**, *15*, 11092. [\[CrossRef\]](#)
24. Redhu, P.; Kumar, K. Short-term traffic flow prediction based on optimized deep learning neural network: PSO-Bi-LSTM. *Phys. A Stat. Mech. Appl.* **2023**, *625*, 129001.
25. Chen, J.; Zheng, L.; Hu, Y.; Wang, W.; Zhang, H.; Hu, X. Traffic flow matrix-based graph neural network with attention mechanism for traffic flow prediction. *Inf. Fusion* **2024**, *104*, 102146. [\[CrossRef\]](#)
26. Xia, D.; Chen, Y.; Zhang, W.; Hu, Y.; Li, Y.; Li, H. RSAB-ConvGRU: A hybrid deep-learning method for traffic flow prediction. *Multimed. Tools Appl.* **2024**, *83*, 20559–20585. [\[CrossRef\]](#)
27. Zhang, X.; Qin, L. An Improved Extreme Learning Machine for Imbalanced Data Classification. *IEEE Access* **2022**, *10*, 8634–8642. [\[CrossRef\]](#)
28. Dong, Z.; Wu, C.; Fu, X.; Wang, F. Research and Application of Back Propagation Neural Network-Based Linear Constrained Optimization Method. *IEEE Access* **2021**, *9*, 126579–126594. [\[CrossRef\]](#)
29. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541), Budapest, Hungary, 25–29 July 2004; IEEE: Piscataway, NJ, USA; Volume 2, pp. 985–990.
30. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [\[CrossRef\]](#)
31. Qi, X.; Yao, J.; Wang, P.; Shi, T.; Zhang, Y.; Zhao, X. Combining weather factors to predict traffic flow: A spatial-temporal fusion graph convolutional network-based deep learning approach. *IET Intell. Transp. Syst.* **2024**, *18*, 528–539. [\[CrossRef\]](#)
32. Luo, Q.; He, S.; Han, X.; Wang, Y.; Li, H. LSTTN: A Long-Short Term Transformer-based spatiotemporal neural network for traffic flow forecasting. *Knowl.-Based Syst.* **2024**, *293*, 111637. [\[CrossRef\]](#)
33. Koçak, Y.; Şiray, G.Ü. New activation functions for single layer feedforward neural network. *Expert Syst. Appl.* **2021**, *164*, 113977. [\[CrossRef\]](#)
34. Thirugnanasambandam, K.; Rajeswari, M.; Bhattacharyya, D.; Kim, J.y. Directed Artificial Bee Colony algorithm with revamped search strategy to solve global numerical optimization problems. *Autom. Softw. Eng.* **2022**, *29*, 1–31. [\[CrossRef\]](#)
35. Jiang, X.; Song, Y.; Xing, L. Dual-Population Artificial Bee Colony Algorithm for Joint Observation Satellite Mission Planning Problem. *IEEE Access* **2022**, *10*, 28911–28921. [\[CrossRef\]](#)
36. Tereshko, V.; Loengarov, A. Collective decision making in honey-bee foraging dynamics. *Comput. Inf. Syst.* **2005**, *9*, 1–7.
37. Li, W.; Xia, L.; Huang, Y.; Mahmoodi, S. An ant colony optimization algorithm with adaptive greedy strategy to optimize path problems. *J. Ambient Intell. Humaniz. Comput.* **2022**, *13*, 1557–1571. [\[CrossRef\]](#)
38. Liu, T.; Yu, Z. The analysis of financial market risk based on machine learning and particle swarm optimization algorithm. *EURASIP J. Wirel. Commun. Netw.* **2022**, *2022*, 1–17. [\[CrossRef\]](#)
39. Zhen, L.; Liu, Y.; Dongsheng, W.; Wei, Z. Parameter estimation of software reliability model and prediction based on hybrid wolf pack algorithm and particle swarm optimization. *IEEE Access* **2020**, *8*, 29354–29369. [\[CrossRef\]](#)
40. Karakoyun, M.; Ozkis, A.; Kodaz, H. A new algorithm based on gray wolf optimizer and shuffled frog leaping algorithm to solve the multi-objective optimization problems. *Appl. Soft Comput.* **2020**, *96*, 106560. [\[CrossRef\]](#)
41. Shariati, M.; Mafipour, M.S.; Ghahremani, B.; Azarhomayun, F.; Ahmadi, M.; Trung, N.T.; Shariati, A. A novel hybrid extreme learning machine-grey wolf optimizer (ELM-GWO) model to predict compressive strength of concrete with partial replacements for cement. *Eng. Comput.* **2020**, *38*, 1–23. [\[CrossRef\]](#)
42. Pandey, A.C.; Kulhari, A.; Shukla, D.S. Enhancing sentiment analysis using Roulette wheel selection based cuckoo search clustering method. *J. Ambient Intell. Humaniz. Comput.* **2022**, *13*, 1–29. [\[CrossRef\]](#)
43. Zeng, T.; Wang, W.; Wang, H.; Cui, Z.; Wang, F.; Wang, Y.; Zhao, J. Artificial bee colony based on adaptive search strategy and random grouping mechanism. *Expert Syst. Appl.* **2022**, *192*, 116332. [\[CrossRef\]](#)

44. Li, X.; Li, S.; Zhou, P.; Chen, G. Forecasting Network Interface Flow Using a Broad Learning System Based on the Sparrow Search Algorithm. *Entropy* **2022**, *24*, 478. [[CrossRef](#)]
45. Wang, Y.; Van Schuppen, J.H.; Vrancken, J. Prediction of traffic flow at the boundary of a motorway network. *IEEE Trans. Intell. Transp. Syst.* **2013**, *15*, 214–227. [[CrossRef](#)]
46. Mallouhy, R.E.; Guyeux, C.; Jaoude, C.A.; Makhoul, A. Forecasting the number of firemen interventions using Exponential Smoothing methods: A case study. In *Proceedings of the International Conference on Advanced Information Networking and Applications*; Springer: Cham, Switzerland, 2022; pp. 579–589.
47. Li, D.; Wang, X.; Sun, J.; Feng, Y. Radial basis function neural network model for dissolved oxygen concentration prediction based on an enhanced clustering algorithm and Adam. *IEEE Access* **2021**, *9*, 44521–44533. [[CrossRef](#)]
48. Xie, Y.; Zhang, Y.; Ye, Z. Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition. *Comput.-Aided Civ. Infrastruct. Eng.* **2007**, *22*, 326–334. [[CrossRef](#)]
49. Wang, C.C.; Li, T.H.; Huang, L.; Chen, X. Prediction of potential miRNA–disease associations based on stacked autoencoder. *Brief. Bioinform.* **2022**, *23*, bbac021. [[CrossRef](#)]
50. Cai, W.; Yang, J.; Yu, Y.; Song, Y.; Zhou, T.; Qin, J. PSO-ELM: A hybrid learning model for short-term traffic flow forecasting. *IEEE Access* **2020**, *8*, 6505–6514. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.