

## Article

# GA-KELM: Genetic-Algorithm-Improved Kernel Extreme Learning Machine for Traffic Flow Forecasting

Wenguang Chai <sup>1</sup>, Yuexin Zheng <sup>1</sup>, Lin Tian <sup>2</sup>, Jing Qin <sup>3</sup>  and Teng Zhou <sup>3,4,\*</sup> 

<sup>1</sup> School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China

<sup>2</sup> Department of Electronics and Engineering, Yili Normal University, Yining 835000, China

<sup>3</sup> Centre for Smart Health, School of Nursing, The Hong Kong Polytechnic University, Hong Kong

<sup>4</sup> School of Cyberspace Security, Hainan University, Haikou 570208, China

\* Correspondence: teng.zhou@polyu.edu.hk

**Abstract:** A prompt and precise estimation of traffic conditions on the scale of a few minutes by analyzing past data is crucial for establishing an effective intelligent traffic management system. Nevertheless, because of the irregularity and nonlinear features of traffic flow data, developing a prediction model with excellent robustness poses a significant obstacle. Therefore, we propose genetic-search-algorithm-improved kernel extreme learning machine, termed GA-KELM, to unleash the potential of improved prediction accuracy and generalization performance. By substituting the inner product with a kernel function, the accuracy of short-term traffic flow forecasting using extreme learning machines is enhanced. The genetic algorithm evades manual traversal of all possible parameters in searching for the optimal solution. The prediction performance of GA-KELM is evaluated on eleven benchmark datasets and compared with several state-of-the-art models. There are four benchmark datasets from the A1, A2, A4, and A8 highways near the ring road of Amsterdam, and the others are D1, D2, D3, D4, D5, D6, and P, close to Heathrow airport on the M25 expressway. On A1, A2, A4, and A8, the RMSEs of the GA-KELM model are 284.67 vehs/h, 193.83 vehs/h, 220.89 vehs/h, and 163.02 vehs/h, respectively, while the MAPEs of the GA-KELM model are 11.67%, 9.83%, 11.31%, and 12.59%, respectively. The results illustrate that the GA-KELM model is obviously superior to state-of-the-art models.

**Keywords:** kernel extreme learning machine; short-term traffic flow forecasting; genetic algorithm

**MSC:** 68Q07; 68Q32; 68W50



**Citation:** Chai, W.; Zheng, Y.; Tian, L.; Qin, J.; Zhou, T. GA-KELM: Genetic-Algorithm-Improved Kernel Extreme Learning Machine for Traffic Flow Forecasting. *Mathematics* **2023**, *11*, 3574. <https://doi.org/10.3390/math11163574>

Academic Editors: Man-Fai Leung, Wenming Cao and Hangjun Che

Received: 30 June 2023

Revised: 14 August 2023

Accepted: 16 August 2023

Published: 18 August 2023



**Copyright:** © 2023 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/>).

## 1. Introduction

Convenient traffic conditions can promote the economic development of a country. However, effective and precise prediction of short-term traffic flows can anticipate upcoming traffic situations based on road traffic data, so to provide drivers with appropriate driving routes to alleviate traffic congestion [1].

Traffic flow exhibits cyclical variations that are often masked by noise or random behavior and influenced by external factors like unforeseen accidents or varying weather conditions [2]. Consequently, it is still a great challenge to propose a scheme for efficient and accurate traffic flow forecasting.

There exists a wide variety of methods and approaches proposed to forecast traffic flow, including non-parametric models and parametric models [3]. Kalman filter (KF) models [4–6], Bayesian vector autoregressive moving [7], time series analysis models [8–10], spectral analysis and statistical volatility models [11,12] and autoregressive integrated moving average (ARIMA) models [13,14] are parametric models. This group of models makes use of designated functions to map the present traffic flow in the future. Nevertheless, models with restricted parameters

struggle to capture the intricate nonlinear relationships within the traffic flow. Consequently, these models often exhibit under-fitting issues.

Non-parametric models encompass support vector regression models (SVR) [15,16], fuzzy logic system methods [17,18], artificial neural network models (ANN) [19,20], deep feature fusion models [21], extreme learning machine (ELM) models [22] and k-nearest neighbour regression models [23,24]. Currently, nonlinear models based on deep learning techniques are of great interest in traffic flow prediction due to their effectiveness in modeling complex nonlinear relationships. Huang et al. [25] proposed a deep belief network architecture with a multitask region, which can learn the effective characteristics of traffic flow prediction in an unsupervised way. Lv et al. [26] applied the stacked automatic encoder model to learn traffic flow features, trained in a greedy layer-wise way. Zhou et al. used a training sample replication strategy to train stacked autoencoders (SAE), thereby improving the accuracy of traffic flow prediction models. Zhou et al. [27] proposed a new multi model ensemble framework based on deep learning for traffic flow prediction. Zhang et al. [28] proposed a new framework for forecasting traffic flow, termed the Spatial-Temporal Graph Diffusion Network (ST-GDN), which learns both the geographical dependence of local regions and global spatial semantics. All the above methods have been confirmed to be effective for predicting traffic flow. Despite their effectiveness, deep learning networks still exhibit some drawbacks when applied in forecasting traffic flow. For example, a deep learning network requires a lot of parameters and takes a long time to learn. Furthermore, because they are based on a gradient descent learning method to iteratively adjust network parameters, they often converge to local minima in practical applications [29]. Therefore, deep learning networks obtain the global optimal value by continuously changing the initial value and running many times.

Huang et al. [30] demonstrated that biases and input weights can be stochastic in the event that the activation function employed within the hidden layer possesses an infinite number of differentiable properties. Accordingly, they proposed extreme learning machine (ELM), an efficient prediction algorithm. ELM has been widely used for forecasting traffic flow [22,31]. Wang et al. [32] proposed improved fuzzy C-means (FCM)-based ELM and demonstrated its superior performance in traffic flow prediction. Compared with the backpropagation algorithm, the ELM algorithm has a superior generalization ability and learns faster [33]. Furthermore, because ELM is based on the principle of the least squares method, it circumvents certain problems of the gradient-based learning method, such as falling into the local minimum and how to select the learning rate. However, ELM requires time to determine the number of hidden layer nodes. In order to eliminate the need to manually adjust the number of hidden nodes and further improve the generalization ability of ELM, kernel extreme learning machine (KELM) was developed by substituting implicit kernel mappings for explicit feature mappings [34]. The selection of parameters is crucial for the performance of the KELM model [35]. Typically, KELM searches for the optimal parameters of the model through the grid search method, which will lead to over-fitting, slow learning speeds, a decrease in generalization performance, etc.

To solve these problems, we propose kernel extreme learning machine optimized by the genetic algorithm [36] in this paper. In this way, we achieve a more accurate prediction performance without increasing the number of parameters, which reduces the problem of over-fitting.

In this paper, the primary contributions are revealed as follows.

- Firstly, we propose a hybrid learning model, termed genetic-algorithm-improved kernel extreme learning machine (GA-KELM), avoiding manual traversal of all possible parameters.
- Secondly, we unleashed the potential prediction accuracy and generalization performance of the kernel extreme learning machine through genetic algorithms.
- Thirdly, this model retains the character of kernel extreme learning machine that has the advantages of rapid learning and a robust generalization ability to deal with non-linear traffic flow through an end-to-end mechanism.

- Fourthly, we have carried out sufficient experiments on GA-KELM and several state-of-the-art traffic flow prediction models on the Amsterdam highway dataset and the England M25 highway dataset and proved the superior performance of GA-KELM.

The subsequent sections of this paper are structured as follows. The Section 2 is the methodology, the Section 3 is the relevant settings for the experiment, and the Section 4 is the discussion based on the results on real-world data. The Section 5 is the conclusions.

## 2. Materials and Methods

This section presents kernel extreme learning machine (KELM) for forecasting traffic flow. Then, we used genetic algorithms to optimize the performance of this model.

### 2.1. Kernel Extreme Learning Machine

Extreme learning machine is a type of machine learning algorithm based on a single-layer feedforward neural network. Compared with the traditional feedforward neural network, the hidden layer parameters in ELM are randomly assigned and then the output weight is calculated according to these parameters, without propagating the error back through the gradient descent algorithm to continuously correct the parameters of the model [31]. ELM exhibits a rapid learning speed, an enhanced fitting capability, and an exceptional generalization performance.

Suppose  $D = \{x_i, y_i\}_{i=1}^M$  is the dataset of  $M$  training samples, where  $x_i = [x_{i1}, x_{i2}, \dots, x_{iE}]^T$  is the input vector and  $y_i = [y_{i1}, y_{i2}, \dots, y_{iF}]^T$  is the output vector. The letters  $E$  and  $F$  represent the quantity of neurons in the input and output layers. The feed-forward neural network with activation function  $o(x)$  and  $l$  hidden nodes can be described as

$$Y = O\beta \tag{1}$$

where  $Y = [y_1, y_2, \dots, y_M]^T$ ,  $O = [o(x_1), o(x_2), \dots, o(x_M)]^T$ , and  $\beta = [\beta_1, \beta_2, \dots, \beta_l]^T$ .  $Y$ ,  $O$ , and  $\beta$  are the target matrix, the output matrix resulting from the hidden layer, and the weight vector connecting the output layer and the hidden layer, respectively.  $o(x_i) = [o(\omega_1^T x_i + b_1), o(\omega_2^T x_i + b_2), \dots, o(\omega_l^T x_i + b_l)]^T, i = 1, \dots, M$ .  $\omega_i, i = 1, \dots, l$  is the weight vector between the input layer and the hidden layer.  $b_i, i = 1, \dots, l$  is the bias of the hidden layer.  $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{iF}]^T, i = 1, \dots, l$  represents the output weight of the  $i$ th node.

To ensure that the estimated value is as close as possible to the real value, the output weight  $\beta$  of ELM can be obtained by minimizing the objective function  $L$ . The objective function of ELM is shown in Formula (2).

$$L = \frac{1}{2} \|O\beta - Y\|^2 + \frac{1}{2} \gamma \|\beta\|^2 \tag{2}$$

The optimal solution of Formula (2) is shown in Formula (3).

$$\beta = (\gamma I + O^T O)^{-1} O^T Y \tag{3}$$

where  $\gamma$  is the regularization factor that balances the influence of the error term and the model complexity.  $I$  represents an identity matrix. Therefore, the output of the new input  $x$  can be shown as

$$f(x) = o(x)\beta \tag{4}$$

After replacing the implicit mapping defined by the kernel with Gaussian kernel  $g(x_1, x_2)$ , the output can be expressed as

$$\begin{aligned}
 f(x) &= \varphi(x)\beta \\
 &= \varphi(x)(\gamma I + A^T A)^{-1} A^T Y \\
 &= \varphi(x) A^T (\gamma I + A A^T)^{-1} Y \\
 &= w(x)\alpha
 \end{aligned}
 \tag{5}$$

where  $g(x_1, x_2) = \exp(-\frac{\|x_1-x_2\|}{2\sigma^2})$ ,  $w(x) = \varphi(x)A^T$ , and  $\alpha = (\gamma I + A A^T)^{-1}Y$ .  $\sigma$  is the kernel size and  $\alpha$  is the coefficient matrix.  $\varphi(x)$  and  $A$  are the activation function and output matrix with the Gaussian kernel, respectively. The kernel trick can be expressed in Equation (6). Then,  $\varphi(x)A^T$  and  $A A^T$  can be, respectively, shown in Formulas (7) and (8).

$$g(x_1, x_2) = \varphi(x_1)\varphi(x_2)^T \tag{6}$$

$$\begin{aligned}
 \varphi(x)A^T &= \varphi(x)[\varphi(x_1)^T, \varphi(x_2)^T, \dots, \varphi(x_M)^T] \\
 &= g(x, x_1), g(x, x_2), \dots, g(x, x_M)
 \end{aligned}
 \tag{7}$$

$$A A^T = \begin{bmatrix} g(x_1, x_1) & \cdots & g(x_1, x_M) \\ \vdots & \ddots & \vdots \\ g(x_M, x_1) & \cdots & g(x_M, x_M) \end{bmatrix}
 \tag{8}$$

The following is the calculation process of KELM. The value of  $\alpha$  is calculated by putting the values of input  $x$ , output  $y$ ,  $\sigma$  and  $\gamma$  into Equation (8). Furthermore, the estimated value of  $f(x)$  is calculated by Formulas (5) and (7).

### 2.2. Genetic Algorithm

The genetic algorithm is an optimization method exploiting biological evolution principles, inspired by Darwin’s theory of evolution and Mendel’s genetic theory, which performs a randomized global search for the optimal solution. In the genetic algorithm, an individual, referred to as a chromosome, represents a solution in the context of this problem. The suitability of the genetic algorithm is assessed using the fitness function, while the group of potential solutions is referred to as the population. The better the adaptability of individuals, the higher the probability that individuals will produce offspring through mating. In the process of mating, the probability of an individual’s crossing and mutation is expressed by the value of the crossing rate and the mutation rate. Over the course of multiple generations, the individual’s fitness value will eventually stabilize at an optimal solution to the problem [36].

The following is the process of the genetic algorithm.

- Step 1. Specify the quantity of iterations and chromosomes and the values of crossover rate and mutation rate.
- Step 2. Generate the chromosomes of the first population  $P$ , where the population  $P = [p_1, p_2, \dots, p_q]$  is the collection of all chromosomes (individuals),  $p_{i=1, \dots, q} = [x_1, x_2, \dots, x_{2k}]$  is the  $i$ th chromosome, whose value is expressed by  $2k$  binary sequences, and  $q$  is the number of individuals. A chromosome is formed by the combination of  $\gamma = [x_1, x_3, \dots, x_{2k-1}]$  and  $\sigma = [x_2, x_4, \dots, x_{2k}]$ .
- Step 3. Map all the individuals  $p_i$  in population  $P$  to a certain range set according to the actual problem.
- Step 4. Calculate the fitness value of each individual by means of the objective function.
- Step 5. The population  $P$  is selected according to the fitness value to reproduce. The greater the individual’s fitness value, the higher the probability of being selected.

- Step 6. The selected population  $P$  breeds offspring and has a certain probability of crossing and mutation.  $m$  and  $c$  stand for the values of the mutation and crossing rates, respectively.
- Step 7. Conduct Step 4 to 6 until the iteration number  $n$  is met.

### 2.3. GA-KELM for Traffic Flow Forecasting

Through the grid search method, we must continuously and manually set plenty of values of  $\gamma$  and  $\sigma$  to train the model in order to discover a group of parameters from these values that can achieve the best training effect for the KELM model. Therefore, to achieve better training results, it is imperative to manually set a vast quantity of parameter values, which will consume a lot of time. In addition, when new samples are added during training, the generalization ability of KELM is insufficient.

This paper puts forward a hybrid model called GA-KELM as a solution to the aforementioned issues. We use the genetic algorithm instead of grid search methods to search for network parameters that are more suitable for KELM. The model iteratively generates the parameters  $\gamma$  and  $\sigma$  of KELM through genetic algorithms. The purpose of crossover is to generate new gene combinations as different as possible from their parents, while mutation avoids local optima. The result of mutation is unstable [37]. The crossover rate is usually set to a value higher than 0.5, while the mutation rate is typically set to a value lower than 0.1. In this experiment, the value of the crossover rate was set from 0.7 to 1. The value of the mutation rate was set between 0 and 0.1. We chose the root mean squared error to construct the fitness function  $f(x)$ . The workflow diagram of GA-KELM is shown in Figure 1. The parameters  $N$ ,  $T$ ,  $m$ , and  $c$  are the number of chromosomes, the maximum iteration number, the value of the mutation rate, and the value of the crossover rate.  $R_1(\gamma_{min}, \gamma_{max})$  and  $R_2(\sigma_{min}, \sigma_{max})$  represent the range of values for the regularization factor  $\gamma$  and the kernel size  $\sigma$ .

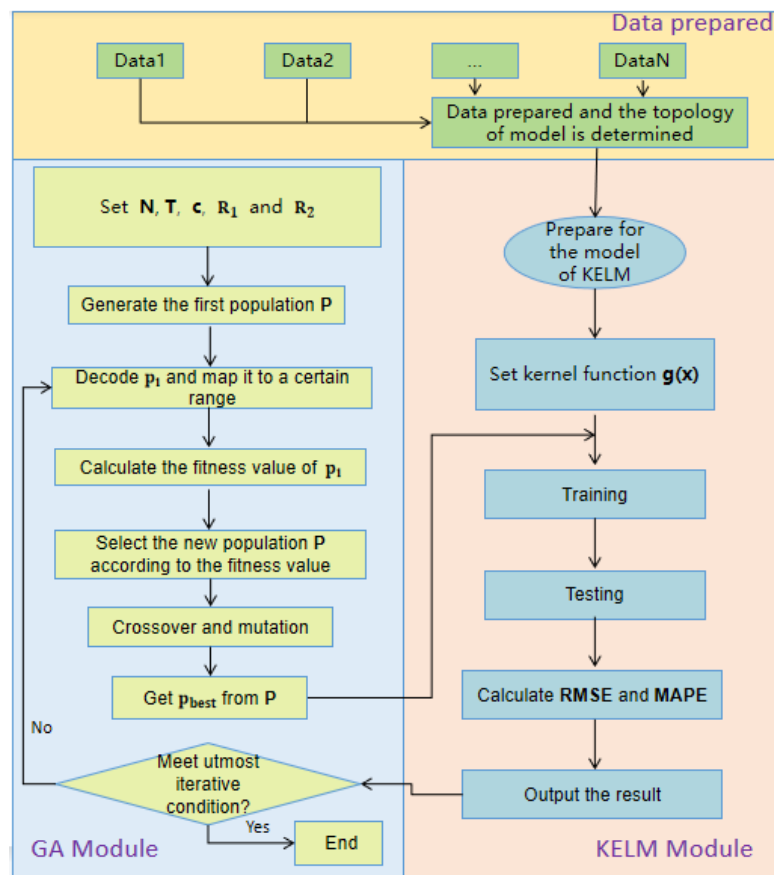


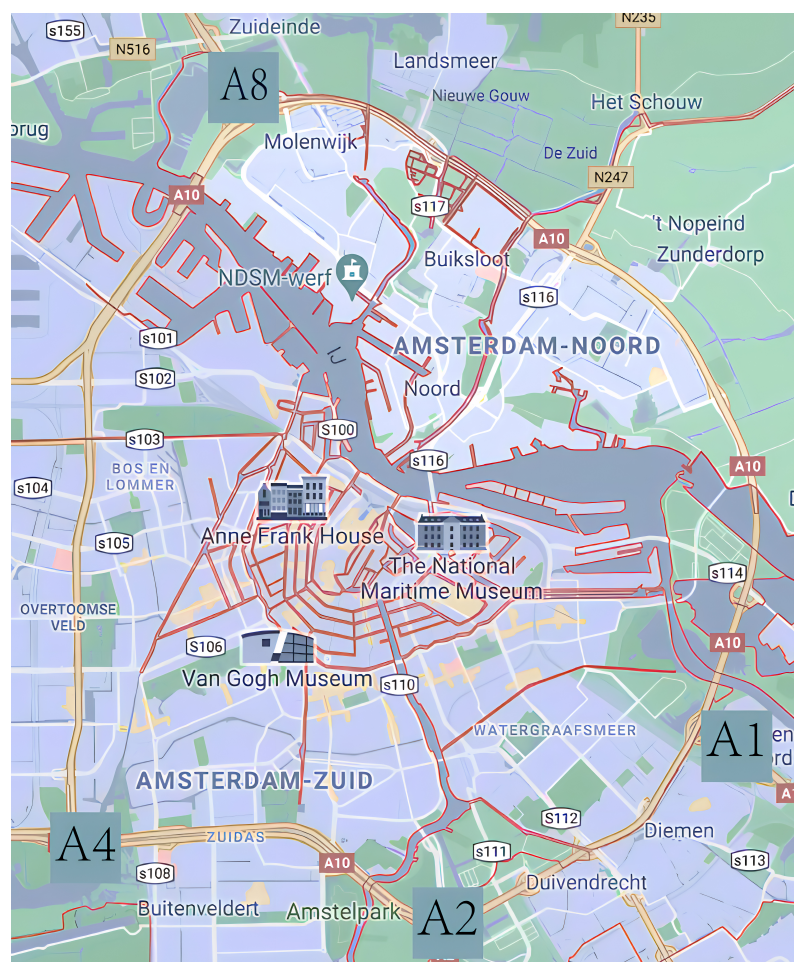
Figure 1. The workflow of the GA-KELM mode.

### 3. Experiments

This section assesses the effectiveness of GA-KELM by analyzing its performance on two benchmark datasets for short-term traffic flow prediction, e.g., the Amsterdam highway dataset and the England M25 highway dataset. More specific details with regard to the case studies are below. The experimental outcomes presented in this study are the average of 50 individual experiments. On A1, A2, A4, and A8, the standard deviations of RMSEs are 0.753, 0.781, 0.897, and 0.794, respectively, and the standard deviations of MAPEs are 0.154, 0.275, 0.163, and 0.161, respectively.

#### 3.1. Datasets Description

The Amsterdam highway dataset comprises real traffic flow data gathered from the A1, A2, A4, and A8 freeways located in the Amsterdam Ring Road [38]. The measurement positions of A1, A2, A4, and A8 are shown in Figure 2. The four benchmark datasets are summarized based on the number of vehicles passing through in each minute per hour. In the experiment, one minute of traffic flow data are summarized as 10 min of traffic flow data.



**Figure 2.** Brief overview of A1, A2, A4, and A8 on the ring road of Amsterdam.

The England M25 highway dataset consists of D1, D2, D3, D4, D5, D6, and P, collected from seven different stations. Figure 3 shows the measurement locations near the intersection of the A10 ring road. Each subset is a collection of the traffic flow for 15 min. There are 2976 data points from 1 August 2019 to 31 August 2019.

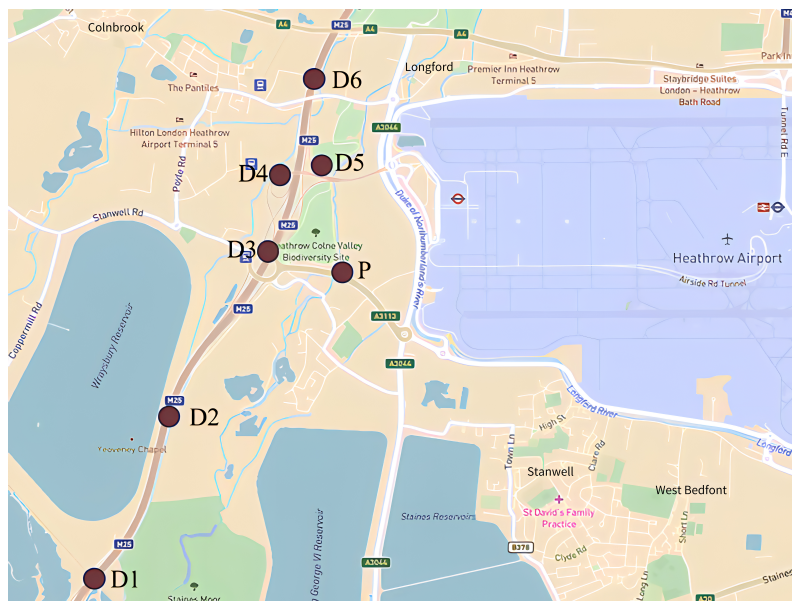


Figure 3. Brief overview of D1, D2, D3, D4, D5, D6, and P on the M25 expressway in England.

### 3.2. Evaluation Criteria

The RMSE calculates the average difference between predicted values and true ones, while the MAPE represents the percentage of the average difference. These two criteria are, respectively, represented by Equations (9) and (10).

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (\bar{y}(n) - y(n))^2} \tag{9}$$

$$MAPE = \sqrt{\frac{1}{N} \sum_{n=1}^N \left| \frac{\bar{y}(n) - y(n)}{y(n)} \right|} \times 100\% \tag{10}$$

where  $y(n)$  and  $\bar{y}(n)$  refer to the actual measurement and prediction values for the  $n$ th data group.

### 3.3. Experimental Setup

In this experiment, we set the data for the initial four weeks for A1, A2, A4, and A8 as the training dataset, and the last week of data as the testing dataset. As for the England M25 highway dataset, the training dataset is composed of data from the initial week, while the testing dataset is composed of data from the second week.

An artificial neural network (ANN) is a computational model that emulates the structure and function of biological neural networks by interconnecting neurons, and is a non-parametric learning model. In this paper, we set the following network parameters according to the standards in [31]. The goal of mean square error is set to 0.001, and there is one hidden layer with a radial basis function (RBF) spread of 2000. The upper limit of neurons in the hidden layer (MN) is set to 40. The default standard number of neurons is set to 25.

Grey prediction (GM) [39] is a method utilized for forecasting systems comprising uncertain factors. GM is based on the past and present development laws of objective things, analyzing future development trends and conditions. The GM(1,1) model was employed to predict traffic flow.

Support vector regression (SVR) is a non-parametric regression model that uses the support vector machine (SVM) algorithm to solve regression problems. For this experiment, we establish the parameters of SVR based on the criteria outlined in [24]. The chosen kernel

type for SVR is the radial basis function (RBF). The cost parameter  $C$  is determined by the maximum deviation allowed between the predicted and actual traffic flow, and the loss function is  $\epsilon$ -insensitive with a value of 1.

The autoregressive (AR) model predicts the current value of a variable by analyzing its past values and establishing a relationship between the historical and current data of the variable. To ensure an appropriate model fit, it is crucial to choose a reasonable value for the order  $p$ , as higher values of  $p$  require the estimation of more parameters. The value of parameter  $p$  is established as 8, identical to that used by Cai et al. [6].

Kalman filtering (KF) is an algorithm that estimates the optimal state of a system by observing its input and output data. To mitigate the impact of noisy data on prediction results and improve the performance of KF, the discrete wavelet decomposition method was used to preprocess the traffic flow data. The criteria outlined in [40] were utilized to establish the relevant parameters.

The seasonal autoregressive integrated moving average (SARIMA) [41] model converts a time series with seasonality into a stationary time series through differential operations. It was specifically designed to leverage the inherent sequential lagged relationships often found in periodically collected data.

The long short-term memory (LSTM) network [42] is a specialized variation of a recurrent neural network (RNN), and is commonly used to predict important events with very long intervals and delays in time series. All parameters of the LSTM network were set as follows. The number of hyper parameter units, the batch size, the epochs, and the validation split were set to 256, 32, 50, and 0.05, respectively.

The noise-immune long short-term memory (NiLSTM) network is a model based on LSTM networks that can eliminate non-Gaussian noise. In the NiLSTM, the cost function is designed based on Correntropy instead of the mean square error (MSE), consequently improving the forecasting performance of the model. NiLSTM is detailed in [43].

A stacked autoencoder (SAE) [44] is a deep learning model that utilizes unsupervised pre-training to enhance model performance. The model conducts unsupervised pre-training for each autoencoder followed by fine-tuning through backpropagation to optimize the performance of each autoencoder.

Extreme learning machine optimized by the genetic algorithm (GA-ELM) is a hybrid learning model utilized for predicting short-term traffic flow. This model employs the genetic algorithm to explore the most effective solution for the extreme learning machine. The values for the maximum number of iterations, generation gap, probability of crossover, and mutation are 100, 0.95, 0.85, and 0.03, respectively, as stated in [31].

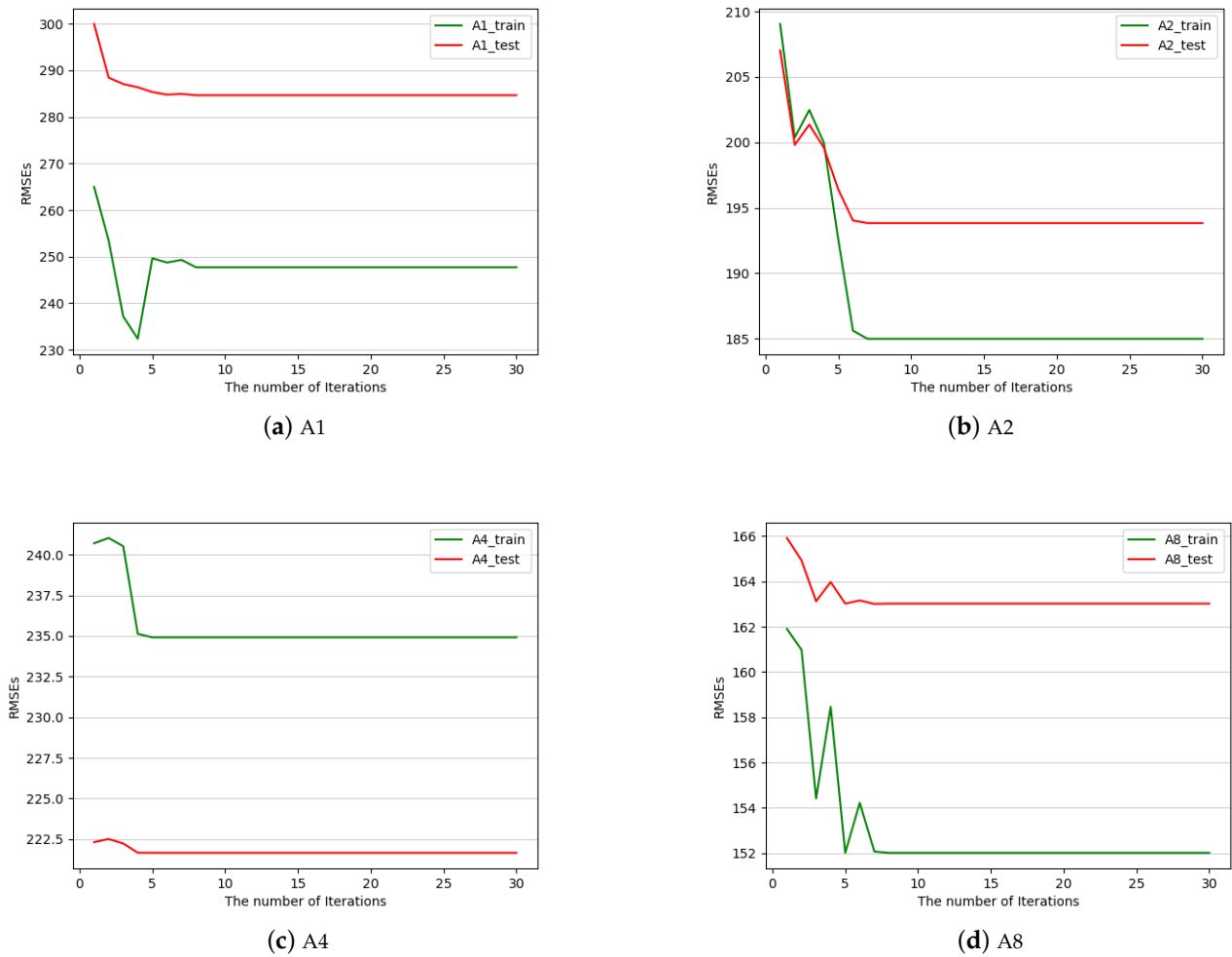
Gravitational search algorithm-optimized extreme learning machine (GSA-ELM) is a blended learning model. Genetic algorithms are utilized to find the optimal solution of the extreme learning machine in the study. The number of hidden layer nodes and the time lag were set to 30 and 8. The maximum number of iterations, the population size, the gravity constant  $G_0$ , and the constant  $v$  were set to 100, 300, 100, and 20, respectively, according to [31].

The study also includes a comparison between GA-KELM and two other methods: the standard extreme learning machine (ELM) and the kernel extreme learning machine (KELM).

#### 4. Results and Discussion

The iteration number of the GA-KELM model is set to 30. The experimental results of GA-KELM on A1, A2, A4, and A8 are shown in Figure 4. The RMSEs for training datasets are displayed using green lines, while the RMSEs for testing datasets are shown using red lines. After surpassing 10 iterations, the fitness function displays negligible fluctuations in value, which proves that the number of iterations is reasonable. In Figure 4a, while the final stable value attained during the training process is not the minimum value, this still does not have any impact on the performance and stability of the model.





**Figure 4.** The RMSE forecasting results with different numbers of GA iterations on A1, A2, A4, and A8.

The GA-KELM model exhibited superior performance across all datasets (A1, A2, A4, and A8) with the lowest RMSE values, as indicated in Table 1. For example, compared to AR, our model performs better on A1, A2, A4, and A8. GA-KELM achieved RMSE values that are lower than those of AR by 5.56%, 9.52%, 2.31%, and 2.21% on the respective datasets A1, A2, A4, and A8. Overall, KELM performs the best on A1 and A2, among several other methods. Compared with KELM, the RMSEs of our model reduce by 0.42%, 2.00%, 0.65%, and 0.42%, respectively. For all models, the RMSEs of A8 are smaller than those of the other three datasets. Due to the different traffic environments on the roads, the traffic flow on the A8 highway is generally smaller than those on the other three highways, making the RMSEs of the A8 dataset the lowest.

In Table 2, the MAPEs of the model proposed by us achieved the best performance on A1, A2, and A4. On the A8 dataset, although our model did not perform the best, the performance of GA-KELM improved compared to KELM. The MAPE represents the proportion of the discrepancy between the predicted and actual values in relation to the actual values, which makes it more sensitive to small measurements or outliers than the RMSE. The traffic flow of the A8 dataset is much smaller than that of other datasets, meaning that the measured values in the A8 dataset are relatively small. Therefore, although the RMSEs of all models on A8 are the smallest, the MAPE is relatively high.

**Table 1.** The RMSEs (vehs/h) of different forecasting models on the Amsterdam ring road dataset.

Models	A1	A2	A4	A8
ANN	299.64	212.95	225.86	166.50
GM	347.94	261.36	275.35	189.57
SVR	329.09	259.74	253.66	190.30
AR	301.44	214.22	226.12	166.71
KF	332.03	239.87	250.51	187.48
SARIMA	308.44	221.08	228.36	169.36
LSTM	294.52	211.31	224.68	168.91
NiLSTM	285.54	203.69	223.72	163.25
SAE	295.43	209.32	226.91	167.01
ELM	294.10	201.67	222.07	169.15
KELM	285.86	197.79	222.34	163.70
GA-ELM	291.42	211.43	228.57	169.25
GSA-ELM	287.89	203.04	221.39	163.24
GA-KELM	<b>284.67</b>	<b>193.83</b>	<b>220.89</b>	<b>163.02</b>

**Table 2.** The MAPEs (%) of different forecasting models on the Amsterdam ring road dataset.

Models	A1	A2	A4	A8
ANN	12.61	10.89	12.49	12.53
GM	12.49	10.90	13.22	12.89
SVR	14.34	12.22	12.23	12.48
AR	13.57	11.59	12.70	12.71
KF	12.46	10.72	12.62	12.63
SARIMA	12.81	11.25	12.05	12.44
LSTM	12.82	11.06	13.71	12.56
NiLSTM	12.00	10.14	11.57	<b>11.76</b>
SAE	11.92	10.23	11.87	12.03
ELM	11.82	10.34	12.05	12.42
KELM	11.76	10.07	11.58	12.61
GA-ELM	11.86	10.30	11.87	12.26
GSA-ELM	11.69	10.25	11.72	12.05
GA-KELM	<b>11.67</b>	<b>9.83</b>	<b>11.31</b>	12.59

The results of the GA-KELM model on the Amsterdam highway dataset are shown in Figure 5. The actual values are represented by a blue line, while the predicted values of GA-KELM are denoted by a red line. The error between the predicted value and the actual value divided by the actual value is expressed by the green line. In Figure 5, it is evident that the relative errors obtained by the GA-KELM model on A1, A2, A4, and A8 tend to approach 0 in most cases, indicating that GA-KELM has great fitting performance on the Amsterdam highway dataset. Additionally, the red and blue lines are extremely close to the A1, A2, A4, and A8 datasets, indicating that the predicted values exhibit remarkable proximity to the actual values, indicating that our model has a good predictive performance. However, during the early morning or late night, there is a significant decrease in the volume of traffic flow, which results in relatively small prediction errors but large relative errors. To minimize the randomness in forecasting performance, we compared our model with the KELM and the GSA-ELM in different periods. The results are depicted in Figure 6. Figure 6a–e displays the predictions of GA-KELM, KELM, and GSA-ELM in some typical scenarios characterized by significant fluctuations in traffic flow, such as during the morning peak period and evening rush hour. The figure demonstrates that GA-KELM achieves a superior prediction accuracy when confronted with uncertainties and variations in traffic flow. Figure 6f displays the predictions of GA-KELM, KELM, and GSA-ELM at midnight, suggesting that our model excels in the low traffic scenario as well.

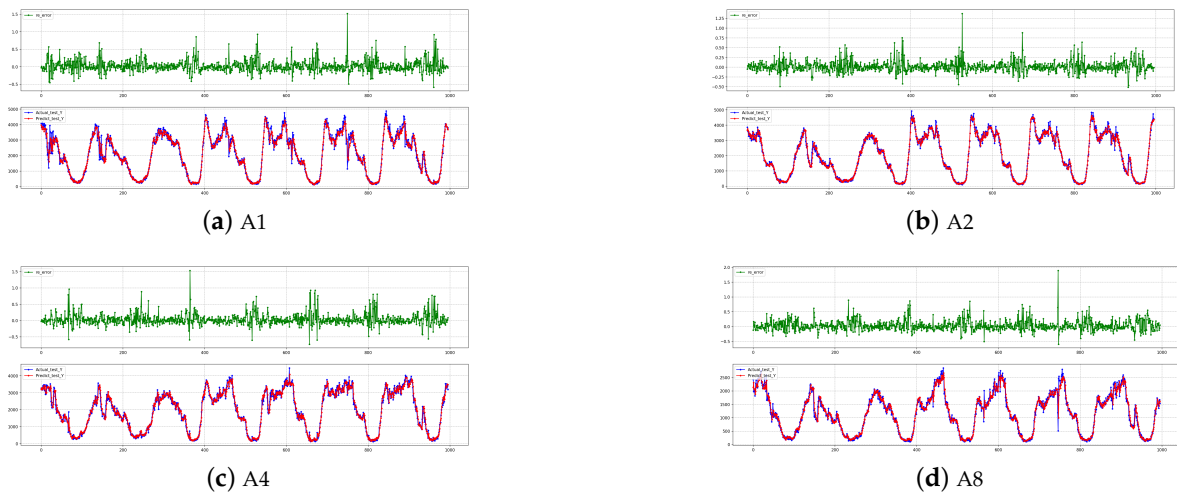


Figure 5. Visualization results of the related error, the measurements, and the predictions of the GA-KELM.

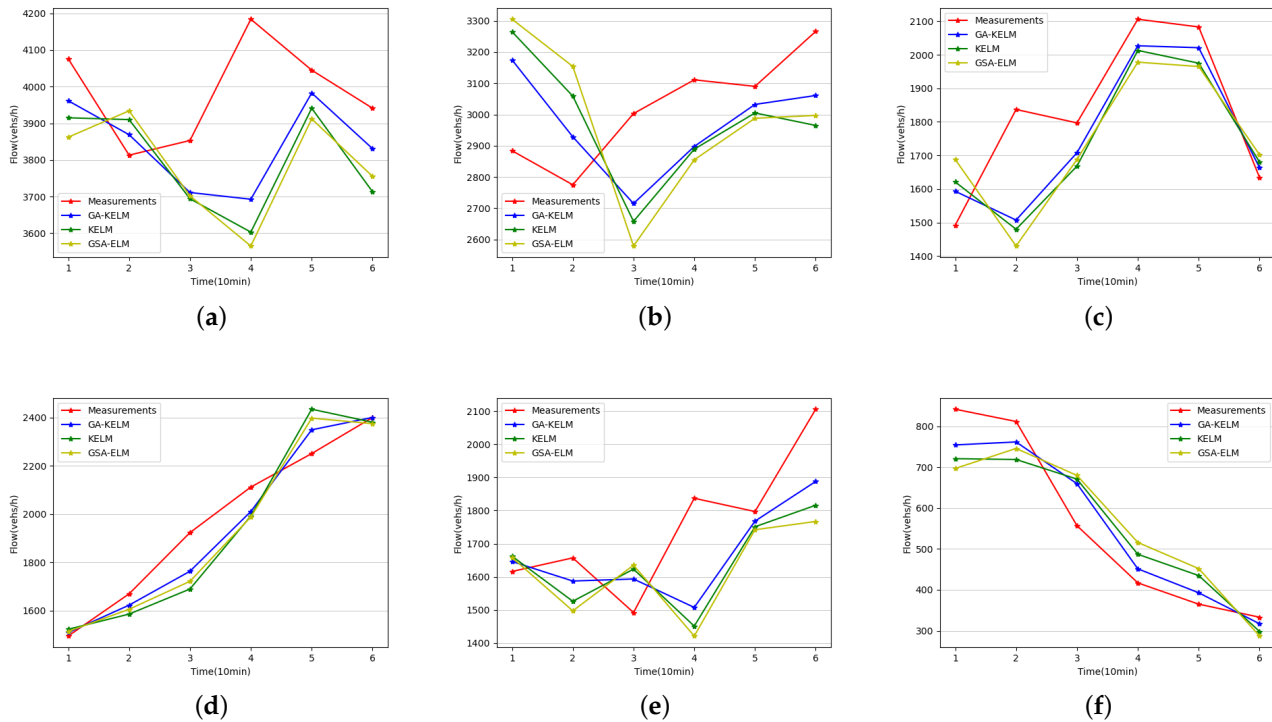


Figure 6. (a–f) display the measurement and the forecasting results of GA-KELM, KELM, and GSA-ELM in different periods.

In Tables 3 and 4, the data from the initial five days, the previous week, the first two weeks, and the first four weeks are set as different sample sizes to train the model, and the last week of data are set as the testing dataset. Our model achieved lower RMSEs and MAPEs than KELM on A1, A2, A4, and A8 with all different sample sizes. For example, compared to KELM, the RMSEs of our model reduced by 2.03%, 3.83%, 1.84%, and 2.09% on A1, A2, A4, and A8, respectively, when the sample size was two weeks. Similarly, the MAPEs reduced by 6.51%, 5.59%, 7.84%, and 5.11%, respectively. Furthermore, we trained our model on A8, A4, A2, and A1 but, respectively, tested it on A1, A2, A4, and A8, represented by A1\*, A2\*, A4\*, and A8\*, respectively. As shown in Tables 3 and 4, the four week training dataset achieves the lowest values of RMSE and MAPE in different sample

sizes. Our model achieved a good performance even when trained on one dataset and tested on another.

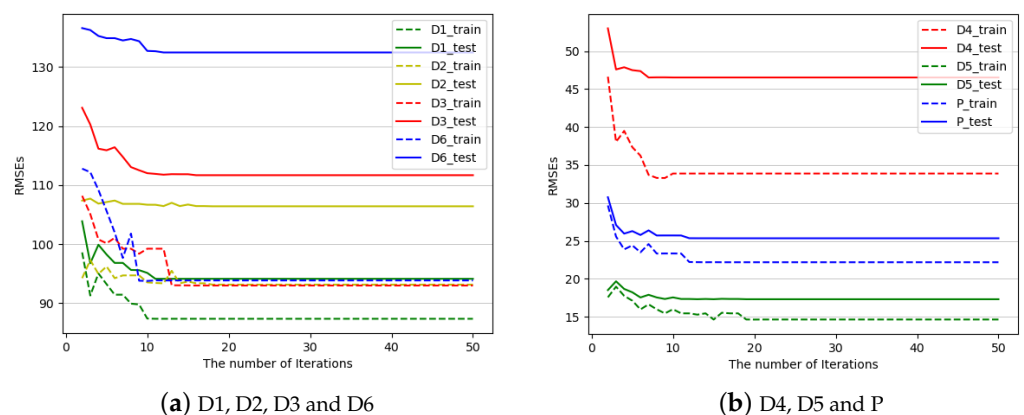
**Table 3.** The RMSE (vehs/h) comparison results on A1, A2, A4, and A8 with different sample sizes of GA-KELM, KELM, and GA-KELM trained on one dataset but tested on another.

Different Sample Sizes	A1 (Ours)	A1 (KELM)	A1* (Ours)	A2 (Ours)	A2 (KELM)	A2* (Ours)	A4 (Ours)	A4 (KELM)	A4* (Ours)	A8 (Ours)	A8 (KELM)	A8* (Ours)
five days	291.87	295.85	305.02	208.28	214.35	213.53	225.81	233.92	232.14	169.82	173.62	166.45
one week	288.22	294.28	306.05	200.71	207.16	208.29	224.74	229.87	231.25	168.85	173.07	167.13
two weeks	288.01	293.97	298.93	198.01	205.89	205.11	223.61	227.81	226.67	165.46	168.99	166.43
four weeks	<b>284.67</b>	<b>285.86</b>	<b>297.52</b>	<b>193.83</b>	<b>197.79</b>	<b>204.56</b>	<b>220.89</b>	<b>222.34</b>	<b>225.36</b>	<b>163.02</b>	<b>163.70</b>	<b>164.66</b>

**Table 4.** The MAPE (%) comparison results on A1, A2, A4, and A8 with different sample sizes of GA-KELM, KELM, and GA-KELM trained on one dataset but tested on another.

Different Sample Sizes	A1 (Ours)	A1 (KELM)	A1* (Ours)	A2 (Ours)	A2 (KELM)	A2* (Ours)	A4 (Ours)	A4 (KELM)	A4* (Ours)	A8 (Ours)	A8 (KELM)	A8* (Ours)
five days	12.39	13.21	13.54	11.21	12.68	13.13	12.17	13.36	12.46	13.01	14.06	13.41
one week	11.78	12.91	13.53	10.05	10.84	12.18	12.26	13.34	12.36	13.13	14.36	13.20
two weeks	11.91	12.74	13.02	10.13	10.73	11.44	11.88	12.89	11.94	12.64	13.32	12.87
four weeks	<b>11.67</b>	<b>11.76</b>	<b>12.89</b>	<b>9.83</b>	<b>10.07</b>	<b>11.30</b>	<b>11.31</b>	<b>11.58</b>	<b>11.85</b>	<b>12.59</b>	<b>12.61</b>	<b>12.69</b>

Figure 7 shows the iterative process of GA-KELM on the English M25 highway dataset. After fifteen iterations, the value of RMSE tends to stabilize. As shown in Table 5, our model exhibited optimal performance on all datasets, achieving 94.12, 106.40, 111.67, 46.53, 17.30, 132.44, and 25.33 on the D1, D2, D3, D4, D5, D6, and P datasets, respectively. The RMSEs of GA-KELM are 7.10%, 3.91%, 6.08%, 5.02%, 6.74%, 8.78%, and 10.11% lower than GA-ELM, respectively. Compared with the standard KELM, the RMSEs of GA-KELM decreased by 0.70%, 0.08%, 1.36%, 2.68%, 1.42%, 4.23%, and 0.55%, respectively. Due to some zero traffic flow in the England M25 highway dataset, it is not appropriate to choose the MAPE as an evaluation indicator. The experiment on the England M25 highway dataset demonstrates that the GA-KELM model has a good generalization ability.



**Figure 7.** The RMSE forecasting results with different GA iterations on D1, D2, D3, D4, D5, D6, and P.

**Table 5.** The RMSE (vehs/h) comparison results of traffic flow datasets from D1 to D6 and P on the England M25 highway dataset.

Models	D1	D2	D3	D4	D5	D6	P
ELM	161.49	116.81	124.22	51.52	19.44	149.33	29.17
GA-ELM	101.31	110.74	118.90	48.99	18.55	145.18	28.18
GSA-ELM	97.47	108.03	113.20	48.87	18.45	140.22	26.02
KELM	94.78	106.49	113.20	47.81	17.55	138.29	25.47
GA-KELM	<b>94.12</b>	<b>106.40</b>	<b>111.67</b>	<b>46.53</b>	<b>17.30</b>	<b>132.44</b>	<b>25.33</b>

## 5. Conclusions

In this paper, a blended learning model is developed for short-term traffic flow forecasting. Kernel extreme learning machine is used to cope with the non-linearity of traffic flow sequences with an end-to-end mechanism. The application of a genetic algorithm facilitates the optimization of the parameters of kernel extreme learning machine. The experimental results on the Amsterdam dataset and the England M25 highway dataset show that the genetic algorithm improves the prediction performance and generalization ability of kernel extreme learning machine. We plan to extend the application of the GA-KELM model to other relevant fields in the future, such as PM2.5 concentration forecasting, investment risk level prediction, product sales forecasting, and so on.

**Author Contributions:** Conceptualization, W.C. and T.Z.; formal analysis, Y.Z. and W.C.; investigation, L.T.; methodology, W.C. and Y.Z.; project administration, J.Q.; resources, W.C. and T.Z.; software, W.C. and Y.Z.; supervision, T.Z.; visualization, L.T. and J.Q.; writing—original draft preparation, W.C., Y.Z. and L.T.; writing—review and editing, Y.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China, grant number 61772143, 61902232; Guangdong Province Key Field R&D Program Project, grant number 2021B0101220006; Natural Science Foundation of Guangdong Province, grant number 2022A1515011590, 2021A1515012302, 2022A1515011978; Key Scientific Research Project of Universities in Guangdong Province, grant number 2020ZDZX3028, 2022ZDZX1007; Guangdong Provincial Science and Technology Plan Project, grant number STKJ202209003.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

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

## References

- Jiang, M.; Liu, Z. Traffic Flow Prediction Based on Dynamic Graph Spatial-Temporal Neural Network. *Mathematics* **2023**, *11*, 2528. [\[CrossRef\]](#)
- Zheng, S.; Zhang, S.; Song, Y.; Lin, Z.; Jiang, D.; Zhou, T. A Noise-Immune Boosting Framework for Short-Term Traffic Flow Forecasting. *Complexity* **2021**, *2021*, 5582974. [\[CrossRef\]](#)
- Liu, H.W.; Wang, Y.T.; Wang, X.K.; Liu, Y.; Liu, Y.; Zhang, X.Y.; Xiao, F. Cloud Model-Based Fuzzy Inference System for Short-Term Traffic Flow Prediction. *Mathematics* **2023**, *11*, 2509. [\[CrossRef\]](#)
- Guo, J.; Huang, W.; Williams, B.M. Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transp. Res. Part C* **2014**, *43*, 50–64. [\[CrossRef\]](#)
- Zhou, T.; Jiang, D.; Lin, Z.; Han, G.; Xu, X.; Qin, J. Hybrid dual Kalman filtering model for short-term traffic flow forecasting. *IET Intell. Transp. Syst.* **2019**, *13*, 1023–1032. [\[CrossRef\]](#)
- Cai, L.; Zhang, Z.; Yang, J.; Yu, Y.; Qin, J. A noise-immune Kalman filter for short-term traffic flow forecasting. *Phys. Stat. Mech. Its Appl.* **2019**, *536*, 122601. [\[CrossRef\]](#)
- Mai, T.; Ghosh, B.; Wilson, S. Multivariate Short-Term Traffic Flow Forecasting Using Bayesian Vector Autoregressive Moving Average Model. In Proceedings of the Transportation Research Board Meeting, Washington, DC, USA, 22–26 January 2012.

8. Poncela, P. Time series analysis by state space methods: J. Durbin and S.J. Koopman, Oxford Statistical Series 24, 2001, Oxford University Press, ISBN 0-19-852354-8, 254 pages, price: £36.00 (hardback). *Int. J. Forecast.* **2004**, *20*, 139–141. [[CrossRef](#)]
9. Ghosh, B.; Basu, B.; O'Mahony, M. Multivariate Short-Term Traffic Flow Forecasting Using Time-Series Analysis. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 246–254. [[CrossRef](#)]
10. Min, W.; Wynter, L. Real-time road traffic prediction with spatio-temporal correlations. *Transp. Res. Part C: Emerg. Technol.* **2011**, *19*, 606–616. [[CrossRef](#)]
11. Zhang, Y.; Zhang, Y.; Haghani, A. A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 65–78. [[CrossRef](#)]
12. Tchrakian, T.T. Real-Time Traffic Flow Forecasting Using Spectral Analysis. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 519–526. [[CrossRef](#)]
13. Moayedi, H.Z. Arima model for network traffic prediction and anomaly detection. In Proceedings of the International Symposium on Information Technology, Kuala Lumpur, Malaysia, 26–29 August 2008.
14. Comert, G.; Bezuglov, A. An Online Change-Point-Based Model for Traffic Parameter Prediction. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1360–1369. [[CrossRef](#)]
15. Hong, W.C.; Pai, P.F.; Yang, S.L.; Theng, R. Highway traffic forecasting by support vector regression model with tabu search algorithms. In Proceedings of the 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, Canada, 16–21 July 2006.
16. Cai, L.; Chen, Q.; Cai, W.; Xu, X.; Zhou, T.; Qin, J. SVRGSA: A hybrid learning based model for short-term traffic flow forecasting. *Intell. Transp. Syst. IET* **2019**, *13*, 1348–1355. [[CrossRef](#)]
17. Zhang, Y.; Zhirui, Y.E. Short-Term Traffic Flow Forecasting Using Fuzzy Logic System Methods. *J. Intell. Transp. Syst.* **2008**, *12*, 102–112. [[CrossRef](#)]
18. Arqub, O.A.; Al-Smadi, M.; Momani, S.; Hayat, T. Application of reproducing kernel algorithm for solving second-order, two-point fuzzy boundary value problems. *Soft Comput.* **2016**, *21*, 7191–7206. [[CrossRef](#)]
19. Zhu, J.Z.; Cao, J.X.; Zhu, Y. Traffic volume forecasting based on radial basis function neural network with the consideration of traffic flows at the adjacent intersections. *Transp. Res. Part C Emerg. Technol.* **2014**, *47*, 139–154. [[CrossRef](#)]
20. Liu, H.; Zuylen, H.; Lint, H.V.; Salomons, M. Predicting Urban Arterial Travel Time with State-Space Neural Networks and Kalman Filters. *Transp. Res. Rec. J. Transp. Res. Board* **2006**, *1968*, 99–108. [[CrossRef](#)]
21. Li, L.; Qu, X.; Zhang, J.; Wang, Y.; Ran, B. Traffic speed prediction for intelligent transportation system based on a deep feature fusion model. *J. Intell. Transp. Syst.* **2019**, *23*, 605–616. [[CrossRef](#)]
22. Cai, W.; Yang, J.; Yu, Y.; Song, Y.; Qin, J. PSO-ELM: A Hybrid Learning Model for Short-Term Traffic Flow Forecasting. *IEEE Access* **2020**, *8*, 6505–6514. [[CrossRef](#)]
23. Zheng, Z.; Su, D. Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm. *Transp. Res. Part Emerg. Technol.* **2014**, *43*, 143–157. [[CrossRef](#)]
24. Cai, L.; Yu, Y.; Zhang, S.; Song, Y.; Zhou, T. A Sample-rebalanced Outlier-rejected k-nearest Neighbour Regression Model for Short-Term Traffic Flow Forecasting. *IEEE Access* **2020**, *8*, 22686–22696. [[CrossRef](#)]
25. 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](#)]
26. 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.* **2015**, *16*, 865–873. [[CrossRef](#)]
27. Zhou, T.; Han, G.; Xu, X.; Han, C.; Huang, Y.; Qin, J. A Learning-Based Multimodel Integrated Framework for Dynamic Traffic Flow Forecasting. *Neural Process. Lett.* **2018**, *49*, 407–430. [[CrossRef](#)]
28. Zhang, X.; Huang, C.; Xu, Y.; Xia, L.; Dai, P.; Bo, L.; Zhang, J.; Zheng, Y. Traffic Flow Forecasting with Spatial-Temporal Graph Diffusion Network. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 15008–15015. [[CrossRef](#)]
29. Fernández-Navarro, F.; Hervás-Martínez, C.; Sanchez-Monedero, J.; Gutiérrez, P. MELM-GRBF: A modified version of the extreme learning machine for generalized radial basis function neural networks. *Neurocomputing* **2011**, *74*, 2502–2510. [[CrossRef](#)]
30. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [[CrossRef](#)]
31. Cui, Z.; Huang, B.; Dou, H.; Tan, G.; Zheng, S.; Zhou, T. GSA-ELM: A hybrid learning model for short-term traffic flow forecasting. *IET Intell. Transp. Syst.* **2022**, *16*, 41–52. [[CrossRef](#)]
32. Xing-Chao, W.; Jian-Ming, H.U.; Wei, L.; Yi, Z. Short-term travel flow prediction method based on FCM-clustering and ELM. *J. Cent. South Univ.* **2017**, *024*, 1344–1350.
33. Wu, K.; Xu, C.; Yan, J.; Wang, F.; Lin, Z.; Zhou, T. Error-distribution-free kernel extreme learning machine for traffic flow forecasting. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106411. [[CrossRef](#)]
34. Huang, G.B.; Zhou, H.; Ding, X.; Rui, Z. Extreme Learning Machine for Regression and Multiclass Classification. *IEEE Trans. Syst. Man Cybern. Part B* **2012**, *42*, 513–529. [[CrossRef](#)] [[PubMed](#)]
35. Shang, Q.; Lin, C.; Yang, Z.; Bing, Q.; Zhou, X. A Hybrid Short-Term Traffic Flow Prediction Model Based on Singular Spectrum Analysis and Kernel Extreme Learning Machine. *PLoS ONE* **2016**, *11*, e0161259. [[CrossRef](#)]
36. Hermawanto, D. Genetic Algorithm for Solving Simple Mathematical Equality Problem. *arXiv* **2017**, arXiv:1308.4675.

37. Godwin Immanuel, D.; Chritober Asir Rajan, C. An Genetic Algorithm approach for reactive power control problem. In Proceedings of the 2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT), Nagercoil, India, 20–21 March 2013; pp. 74–78. [[CrossRef](#)]
38. Wang, Y.; Schuppen, J.; Vrancken, J. Prediction of Traffic Flow at the Boundary of a Motorway Network. *IEEE Trans. Intell. Transp. Syst.* **2013**, *15*, 214–227. [[CrossRef](#)]
39. Guo, H.; Xiao, X.; Jeffrey, F. Urban Road Short-term Traffic Flow Forecasting Based on the Delay and Nonlinear Grey Model. *J. Transp. Syst. Eng. Inf. Technol.* **2013**, *13*, 60–66. [[CrossRef](#)]
40. Zhang, Y.; Ye, Z.; Xie, Y. Short-Term Traffic Volume Forecasting Using Kalman Filter with Discrete Wavelet Decomposition. *Comput. Civ. Infrastruct. Eng.* **2007**, *22*, 326–334
41. Lippi, M.; Bertini, M.; Frasconi, P. Short-Term Traffic Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 871–882. [[CrossRef](#)]
42. Ma, X.; Tao, Z.; Wang, Y.; Yu, H.; Wang, Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 187–197. [[CrossRef](#)]
43. Cai, L.; Lei, M.; Zhang, S.; Yu, Y.; Qin, J. A noise-immune LSTM network for short-term traffic flow forecasting. *Chaos* **2020**, *30*, 1–11. [[CrossRef](#)]
44. Zhou, T.; Han, G.; Xu, X.; Lin, Z.; Han, C.; Huang, Y.; Qin, J.  $\delta$ -agree AdaBoost stacked autoencoder for short-term traffic flow forecasting. *Neurocomputing* **2017**, *247*, 31–38. [[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.