IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 18, 2020, accepted January 26, 2020, date of publication January 29, 2020, date of current version February 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2970250

A Sample-Rebalanced Outlier-Rejected *k*-Nearest Neighbor Regression Model for Short-Term Traffic Flow Forecasting

LINGRU CAI[1,3], YIDAN YU[1], SHUANGYI ZHANG[1], YOUYI SONG[2], ZHI XIONG[1,3], AND TENG ZHOU[1,2,3]

[1]Department of Computer Science, College of Engineering, Shantou University, Shantou 515063, China
[2]Center for Smart Health, School of Nursing, The Hong Kong Polytechnic University, Hong Kong
[3]Key Laboratory of Intelligent Manufacturing Technology, Ministry of Education, Shantou University, Shantou 515063, China

Corresponding author: Teng Zhou (zhouteng@stu.edu.cn)

This work was supported in part by the Natural Science Foundation of Guangdong Province under Grant 2018A030313889 and Grant 2018A030313291, in part by the Natural Science Foundation of China under Grant 61902232 and Grant 61902231, in part by the Education Science Planning Project of Guangdong Province under Grant 2018GXJK048, and in part by the STU Scientific Research Foundation for Talents under Grant NTF18006.

ABSTRACT Short-term traffic flow forecasting is a fundamental and challenging task due to the stochastic dynamics of the traffic flow, which is often imbalanced and noisy. This paper presents a sample-rebalanced and outlier-rejected *k*-nearest neighbor regression model for short-term traffic flow forecasting. In this model, we adopt a new metric for the evolutionary traffic flow patterns, and reconstruct balanced training sets by relative transformation to tackle the imbalance issue. Then, we design a hybrid model that considers both local and global information to address the limited size of the training samples. We employ four real-world benchmark datasets often used in such tasks to evaluate our model. Experimental results show that our model outperforms state-of-the-art parametric and non-parametric models.

INDEX TERMS Intelligent transportation systems, road transportation, time series analysis, stochastic processes.

I. INTRODUCTION

Timely and accurate short-term traffic flow forecasting is of crucial importance for proactive traffic management and control systems [1], which can not only subsequently alleviate traffic congestion and reduce carbon emissions, but also ensures the efficiency of traffic operation [2]. It also gains more and more attention to many traffic-related applications, such as vehicle navigation [3], route planning [4], and traffic control [5], etc.

Over the last few decades, a variety of approaches have been proposed [2]. The existing theories and approaches can be roughly divided into two categories, parametric and non-parametric ones. Parametric approaches explicitly and quantitatively formulate the relationship between the input and the output via a function (model), whose parameters need to be estimated, whereas the later ones explore the implicit relationship between the prediction and input data without providing any well-defined functions [6]. The traditional parametric methods include seasonal mean models [7], smoothing techniques [8], [9], time-series models [10], [11], Kalman filtering methods [12]–[14], etc. This type of methods are explicit and easy to understand, but they require specific domain knowledge or expertise to be functioned well [15]. However, improper assumptions or simplifications may degrade the forecasting accuracy of such models in practical applications, especially for real-time operation in intelligent transportation systems with complex and large-scale calculations. Meanwhile, the emerging sensor and storage technologies enable the successful deployment of non-parametric methods. For recent studies, higher forecasting accuracy achieved by non-parametric models are reported, such as *k*-nearest neighbors algorithm (*k*NN) [16], [17], support vector machine (SVM) [18]–[20], extreme learning machine (ELM) [21], and artificial neural networks (ANN) [22], [23], etc.

The associate editor coordinating the review of this manuscript and approving it for publication was Baozhen Yao.

In recent years, deep learning techniques have attracted widely academic and industrial attentions [24], which are able to discover the implicit non-linear relationships inside the traffic flow data using a general purpose learning procedure automatically. Deep learning techniques have also been proven promising for traffic flow forecasting [2], [25], [26]. Lv *et al.* [2] applied a stacked autoencoder (SAE) network for traffic flow forecasting. Zhou *et al.* [25] found that a single SAE with fixed parameters can hardly handle various traffic conditions. For further improving, they proposed a δ-agree AdaBoost strategy to integrate a series of stacked autoencoders for better forecasting. Then, Zhou *et al.* [26] continuously improved the forecasting performance by a deep learning framework that integrates heterogeneous forecasting models. The empirical studies have proven that deep learning models can achieve superior accuracy. However, the success of deep learning approaches require huge amount of data, tedious training time and large computing source, which are not suitable for real-time traffic flow forecasting with limited samples due to the incomplete traffic infrastructure or the bandwidth of the sensor network [20]. Furthermore, the duration of different traffic flow patterns, such as peak hours and off-peak hours, are different within a day and change dramatically from day to day. Such a phenomenon leads to the imbalance of training samples, and subsequently affects the performance of deep learning-based approaches. Furthermore, traffic flow not only exhibits periodic variations obscured by noises, but also reveals stochastic behaviors affected by external factors, such as extreme weather or unexpected incident [27].

To address these issues, we rethink the potential improvement of k-nearest neighbor (kNN) model. The kNN model tackles the nonlinear problem in an intuitive, self-learning, and effective routine [28], and has been widely applied in various areas, such as traffic flow forecasting [29]–[31]. For example, Li *et al.* [29] proposed a k-nearest neighbor locally weighted regression method for short-term traffic flow forecasting. Hong *et al.* [30] proposed a hybrid multi-metric based k-nearest neighbor method to resize the intrinsic features of the traffic flow data for traffic flow forecasting. However, the conventional kNN based short-term traffic flow forecasting approaches are easily degraded in some real-world situations. For example, the algorithm may be executed under limited training samples, imbalance size of training samples of daily traffic flow, or noisy or outlier samples caused by hardware failure, etc. Thus, the performance of conventional kNN based traffic flow forecasting approaches is highly depended on the quantity and quality of the training samples. Furthermore, the seasonality and the trend of the traffic flow evolves over time. Outlier detection and rejection are also an important research topic and many effective methods have been proposed [32]–[34]. One of the important branches is to use clustering techniques to detect outliers, such as fuzzy clustering [35], [36], and gaussian measure model [37]. The Euclidean distance metric frequently used by conventional kNN can only describe the overall degree of similarity for two sequences with equal length. The local information is obscured and the similar traffic flow patterns with unequal length are often ignored. Thus, a proper similarity metric that considers the variation of different traffic flow patterns requires further investigation.

Aiming at the three aforementioned issues, we propose a sample-rebalanced outlier-rejection k-nearest neighbor regression ($\mathrm{S}^r\mathrm{O}^r k\mathrm{NN}^r$) model for short-term traffic flow forecasting. Our $\mathrm{S}^r\mathrm{O}^r k\mathrm{NN}^r$ not only inherits the advantages of traditional kNN, but is also robust to noisy and imbalanced traffic flow. Note that although the proposed model is applied to traffic flow forecasting in this study, it is general, and can be easily extended to other prediction tasks, such as the image contour prediction [38], grid load demand forecasting [39], or forecasting demand in a shared bicycle or taxi system [40].

We summarize the major contributions of this work below:

- First, we explore the regularities and potential pattern of the traffic flow by adopting the dynamic time warping (DTW) distance metric, which is designed to align temporal sequences and more effective to measure similarities in high-dimension feature space. Then, we reconstruct a new balance training set by relative transformation to tackle the imbalance issue of the traffic flow datasets.
- Second, we propose a sample-rebalanced outlier-rejection k-nearest neighbor regression model that integrates k-local hyperplane distance nearest neighbor and fuzzy kNN to fuse the local and global information to address the sparse problem caused by the small size of training samples. Two information weights are designed to balance two kinds of environments. Furthermore, we embed a random subspace framework in the final prediction model to further improve the prediction performance by reducing the interference of noises.
- Third, we evaluate our model on four real-world benchmark datasets collected from the four highways of Amsterdam by comparing it with several state-of-the-art methods. The results demonstrate the superior performances of our method.

The remaining of the paper is organized as follows. The second part introduces the DTW metric algorithm and the traditional kNN model. The third presents our sample-rebalanced outlier-rejection k-nearest neighbor regression ($\mathrm{S}^r\mathrm{O}^r k\mathrm{NN}^r$) model. The fourth part is the empirical study of the real-world data from Amsterdam, Netherlands. Finally, the last part summarizes the main findings and discusses future research.

II. PRELIMINARIES

A. DYNAMIC TIME WARPING

A lot of similarity metrics have been proposed in literature [41], [42]. The conventional k-nearest neighbors models for short-term traffic flow forecasting are often equipped with the Euclidean distance to measure the similarity between the traffic flow patterns and the query pattern. However, the Euclidean distance can only describe the

roughly global similarity of two patterns, and the local information is obscured [43]. Dynamic time warping (DTW) is a metric for measuring the similarity between two temporal sequences by aligning temporal sequences in time series [43]. DTW allows temporal sequences to be locally shifted, contracted and stretched, and it could calculate a global optimal alignment path between two given sequences under certain restrictions.

DTW is a typical optimization problem solved by dynamic programming. It uses a specific time warping function $\mathcal{W}$ to describe the time corresponding relationship between the query sample and the referenced sample. Then, we solve the regular function corresponding to the minimum cumulative distance when the two samples match. Given two time series $Q = (q_1, q_2, \ldots, q_n)^\top$ and $C = (c_1, c_2, \ldots, c_m)^\top$, whose lengths are n and m, respectively. We construct a n-by-m matrix grid $d(Q, C) \in \mathbb{R}^{n\times m}$. The (i, j)th element represents the distance $d(q_i, c_j)$ between the point q_i and the point c_j, which means the similarity between each point of the sequence Q and each point of C. A series of potential distance metrics are possible. One of most widely used one is the squared Euclidean distance $d(i, j) = \|q_i - c_j\|_2^2$.

Then we find out a warping path $\mathcal{W}$, which is a sequence of distance points, to describe the alignment of the elements of Q and C, where each w_k corresponding to a point $(i, j)_k$.

$$w_k \in \mathcal{W}. \tag{1}$$

After we have defined the distance measure, the dynamic time warping problem is defined as a global minimization over potential warping paths based on the cumulative distance of each path.

$$D(Q, C) = \min_{\mathcal{W}} \left[\sum_{k=1}^{p} d(w_k) \right]. \tag{2}$$

Then, we restricted the space of possible warping paths to satisfy the boundary monotonicity and step-pattern conditions in the dynamic programming formulation. In this way, we find out two sequences of the same length l, indexed α and β. The element of index $\alpha(i)$ in time series Q matches the element of index $\beta(i)$ in time series C.

$$\alpha(1) = \beta(1), \tag{3}$$
$$\alpha(l) = n, \tag{4}$$
$$\beta(l) = m. \tag{5}$$
$$\alpha(1) \le \alpha(a) \le \cdots \le \alpha(l), \tag{6}$$
$$\beta(1) \le \beta(2) \le \cdots \le \beta(l). \tag{7}$$
$$(\alpha(i+1), \beta(i+1)) - (\alpha(i), \beta(i)) \in (1, 0), (1, 1), (0, 1) \tag{8}$$

We search the optimal alignment path $\mathcal{W}^*$, which satisfy the above restriction, by the following recursive formula.

$$\gamma(i, j) = d(i, j) + \min\{\gamma(i-1, j), \gamma(i-1, j-1), \gamma(i, j-1)\} \tag{9}$$

where $\gamma(i, j)$ is the cumulative distance, e.g. the sum of the distance between current elements and the minimum of the cumulative distances of the neighbouring points. In this work, we employ the squared Euclidean distance to compute $d(i, j)$. Then, we will improve the prediction of the DTW based kNN regression in the following part.

B. CONVENTIONAL kNN REGRESSION

The target of short-term traffic flow forecasting task is to predict the future traffic flow based on the historical traffic flow and other extra information. Without loss of generality, we define the short-time traffic flow state vector x at the current moment t as following:

$$x_t = [v_{t-h+1}, \ldots, v_t], \tag{10}$$

where the traffic flow x_t denotes the traffic flow state vector of the road section at current time t, v_t denote the traffic flow of the road section at the moment t.

The conventional kNN regression for traffic flow forecasting contains two stages. First, after constructing the traffic flow state vector, we select k nearest neighbor traffic flow state vector by a certain metric to measure the similarity between the current traffic flow state and the historical traffic flow state vector. Conventional kNN regression often adopts Euclidean distance as the distance metric. However, the Euclidean distance is not always the optimal distance metric for short-term traffic flow forecasting task, since traffic flow is easily affected by noises and outliers.

Second, we can predict the traffic flow by fusing the weighted results of the selected neighbor traffic flow state vectors according to the similarity between the current traffic flow state vector and the historical traffic flow state vectors.

Although the traditional kNN algorithm has been widely used in short-term traffic flow forecasting and achieved good performance, the imbalance and noise statistic characteristics inside the traffic flow data still can not be address.

III. METHODOLOGY

In this section, we elaborate on our sample-rebalanced outlier-rejected k-nearest neighbor regression model for short-time traffic flow forecasting. The overview of our $\mathrm{S}^r\mathrm{O}^r k\mathrm{NN}^r$ model is shown in Fig. 1. In summary, our $\mathrm{S}^r\mathrm{O}^r k\mathrm{NN}^r$ model contains five stages. We firstly pad the missing data of the original traffic flow sequences by averaging the adjacent traffic flow rate, and then cluster the padded data. Second, our model generates the balance training dataset by relative transformation strategy, which is employed to tackle the imbalance and noisy of the dataset. Third, our $\mathrm{S}^r\mathrm{O}^r k\mathrm{NN}^r$ model constructs the local hyperplane of the traffic flow data for each cluster, and then calculates the distances of the local hyperplanes. Fourth, we design two kinds of probabilities corresponding to the distance of the hyperplane of kth local nearest neighbor (HKNN) and the kth fuzzy nearest neighbor, respectively. These two kinds of probabilities represent the local and global information of the training samples. Lastly, we develop a random subspace ensemble framework that generates multiple random

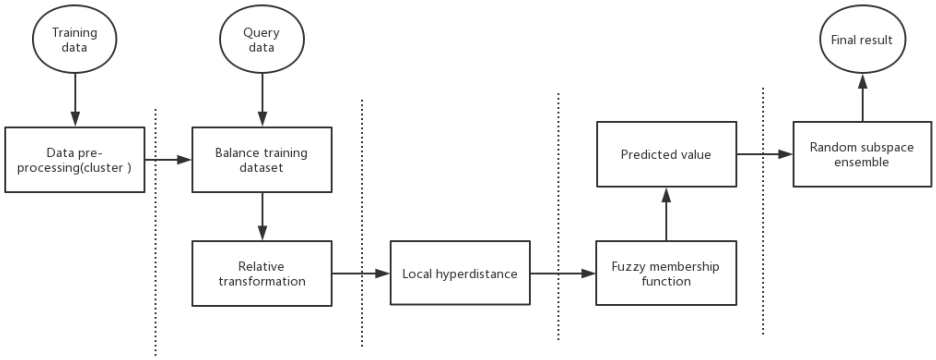

FIGURE 1. The overview of our $S^rO^rkNN^r$ model.

subspaces to improve the performance of the final forecasting accuracy.

A. DATA PREPROCESSING AND RELATIVE TRANSFORMATION

The conventional kNN for short-term traffic flow prediction have their drawbacks. The potential patterns of the traffic flow data tend to be diverse, since the stochastic dynamic and nonlinearity of the traffic flow. Thus, the performances of the kNN based forecasting algorithms are highly depended on the representative and extensive training data selected. Furthermore, Euclidean distance embedded in the conventional kNN forecasting algorithms is sensitive to noisy outliers. To address this problem, we try to cluster traffic flow data to find different categories of traffic flow data with the same pattern in data pre-processing using DTW as the similarity measure algorithm.

To facilitate the following discussion, we introduce some additional notations below. The group of traffic flow data S of m points $\{x_1, x_2, \ldots, x_m\}$, $x_i \in \mathbb{R}^n$, where x_i is the traffic flow state vector. Specifically, x_i can be denoted as $\{x_{i_1}, x_{i_2}, \ldots, x_{i_j}\}$, and the traffic flow value of next interval moment is given as the true traffic flow value y_i corresponding to x_i. After clustering, $H = \{h_1, h_2, \ldots, h_m\}$ is the corresponding set of the labels to traffic flow dataset S, where $h_i \in C$, $C = \{C_1, C_2, \ldots, C_j\}$ is the set of classes. Each class is denoted as C_j, where j is the index of the class.

Assuming that the query traffic flow state vector q is the current traffic flow state vector that does not belong to the training dataset. Our task is to give the prediction $\hat{y}$ corresponding to the query state vector q. k nearest neighbors are selected by kNN algorithm in a local hyperplane. To construct the local hyperplane, we first construct a local environment for the query vector q, termed new balanced training set S'. We define the new balanced training set as:

$$\begin{aligned} S' &= \cup_j S_j(\epsilon, q), \\ S_j(\epsilon, v) &= \left\{ x_i \in C_j | d(x_i, q) \leq d_j^{\epsilon} \right\}. \end{aligned} \tag{11}$$

In Eq. 11, ϵ is the neighbourhood size of the query vector q for each class C_j based on the DTW distance, where ϵ is an integer All x_i close to the query vector q compose the local environment of q, which avoid the imbalance of the training set by selecting the feature vectors from each class with equal probability.

In the second step, $S^rO^rkNN^r$ reduces the effects of noisy data by adopting relative transformation [28] to construct a new relative feature space. The new training set S'' be generated as follows:

$$S'' = S' \cup q. \tag{12}$$

In Eq. 12, the query vector q is also included in the balanced training set constructed in the previous step. Then, we denote x as the feature vector in the new training set S''. The distances between x and each feature x_i in S'' are calculated to make up a set of distance values, which is denoted as follows:

$$\{d(x_1, x), \ldots, d(x_{n'-1}, x), d(q, x)\} \tag{13}$$

After such relative transformation, the feature vectors close to each other vector will be closer in the new relative feature space, whereas the feature vectors far from each other will be farther in the relative space.

B. k-LOCAL HYPERPLANE DISTANCE NEAREST NEIGHBOUR ALGORITHM

k-local hyperplane distance nearest neighbor algorithm (HKNN) is an effective model widely used various applications, which solves the imbalance issue and the sparse issue of the dataset at the same time. The intuition of HKNN to handle the case of limited samples is treating the *missing* samples as *holes* and introducing artefacts in the decision surface produced by the training samples [44]. The missing points are fantasized based on a local linear approximation of the manifold of each class [44]. We use the HKNN to calculate the k-local hyperplane distances of the query vector q for all classes. k is the neighborhood size in each class. We define a local hyperplane for each class C_j with k feature vectors, which are k nearest neighbors selected for the tested vector q from each class C_j:

$$H_j^k(q) = \left\{ p | p = \bar{x} + \sum_{t=1}^{k} \alpha_t (x_t - \bar{x}) \right\}, \tag{14}$$

$$s.t. \alpha_{1..k} \in \mathbf{R}^n,$$

$$\bar{x} = \frac{1}{k} \sum_{t=1}^{k} x_i. \tag{15}$$

The k neighborhood $X_j^k(q)$ of a query point q is the set of the k points of S' whose distance to q is smallest, is defined $X_j^k(q) = \{x_1, x_2, \ldots, x_k\}$. Then, the hyperplane distance $d(q, H_j^k(q))$ between the query vector q and the jth local hyperplane $H_j^k(q)$ be computed as following:

$$d(q, H_j^k(q)) = \min_{p \in H_j^k(q)} \|q - p\|, \\ = \min_{\alpha_t} \left\| q - \bar{x} - \sum_{t=1}^{k} \alpha_t (x_t - \bar{x}) \right\|. \quad (16)$$

where α_i can be solved through solving a linear system as follows:

$$(V' \cdot V) \cdot A = V' \cdot (q - \bar{x}), \quad (17)$$

where $A = (\alpha_1, \alpha_2, \ldots, \alpha_k)'$, and V is an $n \times k$ matrix whose columns are the vectors $V_t = x_t - \bar{x}$.

To penalize the large values α_t, a penalty term λ is introduced to the formula. So the k-local hyperplane distance can be redefined as:

$$d(q, H_j^k(q)) = \min_{\alpha_t} \left\{ \left\| q - \bar{x} - \sum_{t=1}^{k} \alpha_t (x_t - \bar{x}) \right\| + \lambda \sum_{t=1}^{k} \alpha_t^2 \right\}. \quad (18)$$

C. SAMPLE-REBALANCED OUTLIER-REJECTED k-NEAREST NEIGHBOUR REGRESSION

Our S^{r}O$^r k$NN^r summarizes the local hyperplane distances by adopting the fuzzy membership function to calculate the fuzzy value m_j as the following formula for each class C_j in the fourth step. The fuzzy value m_j contains the information about the local information of the query vector q, whereas the conventional kNN algorithm only considers the global information of the query vector. According to the fuzzy membership value obtained by the local information of the query vector in the previous step, we further generate a probability value p_j^1 of the query vector q for each class C_j. The detailed calculation process is as follows:

$$m_j = \frac{1}{\sum_{h=1}^{k} \left(\frac{d(q, H_j^k(q))}{d(q, H_h^k(q))} \right)^{\frac{2}{f-1}}}, \quad (19)$$

$$p_j^1 = \frac{m_j}{\sum_{h=1}^{k} m_h}, \quad (20)$$

where $\frac{2}{f-1}$ is the fuzziness exponent, $d(q, H_j^k(q))$ denotes the Euclidean distance between the local hyperplane $H_j^k(q)$ and the tested vector q when f be set to 2. The probability p_j^1 represents the confidence of the prediction.

In order to take the local information and the global information into account, our S^{r}O$^r k$NN^r obtains another fuzzy membership by using the conventional fuzzy kNN. In general, fuzzy kNN selects a neighbor set N that consists of κ nearest neighbors of q in training dataset.

We assume that the neighbour set $N = \{x_1, x_2, \ldots, f_\kappa\}$. Similar to the calculation method of the previous fuzzy membership value m_j, the fuzzy membership value m_i ($i \in \{1, \ldots, \kappa\}$) is calculated as follows:

$$m_i = \frac{1}{\sum_{h=1}^{\kappa} \left(\frac{d(q, x_i)}{d(q, x_h)} \right)^{\frac{2}{f-1}}}, \quad (21)$$

where $d(q, x_i)$ is the Euclidean distance between the neighbor vector x_i and the query vector q when f be set to 2.

Then, the probability p_j^2 that represents the confidence of the prediction of the query vector q provided by each class C_j can be computed as follows:

$$p_j^2 = \frac{M_j}{M} \\ = \frac{\sum_{i=1}^{\kappa} m_i}{\sum_{x_i \in N_j} m_i}, \quad (22)$$

where N_j is the subset of the neighbor set N that belong to the class C_j. The total fuzzy membership value M is the sum of all m_i. M_j is the sum of the m_i whose feature vector x_i belongs to class C_j.

We introduce two parameters ω_1 and ω_2 to summarize the probability values p_j^1 and p_j^2 of each class C_j as follows:

$$\lambda_j = \omega_1 \times p_j^1 + \omega_2 \times p_j^2 \quad (23)$$

Notice that if ω_1 and ω_2 are set to 0 and 1, respectively, S^{r}O$^r k$NN^r will be HKNN. If ω_1 and ω_2 are set to 1 and 0, respectively, S^{r}O$^r k$NN^r will be conventional kNN. The appropriate value of ω_1 and ω_2 can be balanced in two different parts to achieve better performance. The final prediction of the traffic flow value $\hat{y}$ of the query vector q can be computed by S^{r}O$^r k$NN^r according to the value of λ as follows:

$$\hat{y} = \sum_{j=1}^{C} \lambda_j \left(\frac{1}{k} \sum_{t} k x_t \right) \quad (24)$$

IV. EXPERIMENTS

In this section, four real world traffic datasets are used to evaluate the proposed sample-rebalanced outlier-rejection k-nearest neighbor regression model with competing methods. The traffic flow data collected from the four highways A1, A2, A4, and A8 which end at Amsterdam Ring Road (the A10 highway), as shown in Figure 2.

A. DATASET

The dataset used to evaluate the models for short term traffic flow prediction was collected by Wang et al. [45], which includes the traffic flow data of four motorways ending on the ring road of Amsterdam, namely A1, A2, A4, and A8. As shown in Fig. 2, four measurement sites located on the motorways were indicated by yellow curve before the distance from the merge points to the ring road. The traffic flow data were collected by MONICA sensors from May 20, 2010 until June 24, 2010, which are detected on real time and uploaded in 1-min aggregation. The raw data over five weeks

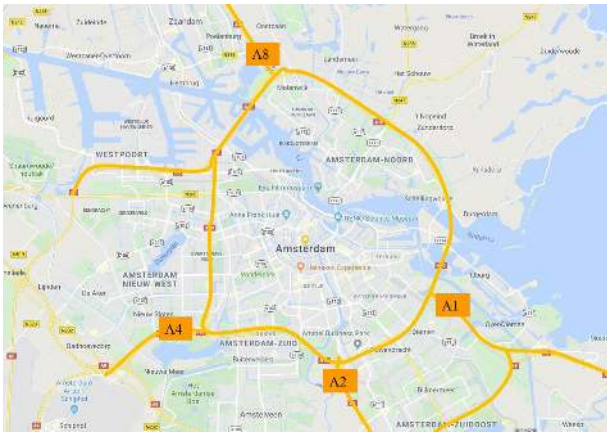

FIGURE 2. Four highways, namely A1, A2, A4, and A8 end at A10 ring road in Amsterdam.

and contains about 20,000 instances. The four highways are described specifically as follows:

The special location of the A1 motorway, connecting the city of Amsterdam with the German border, is a very important route in Europe. There is the first high-occupancy vehicle (HOV) 3+ barrier-separated lane in Europe on A1 motorway, which causing the traffic flow in this HOV lane dramatically changes over time and making the forecasting quite challenging.

The A2 motorway is one of the busiest highways in the Netherlands, which connects the city of Amsterdam and the Belgian border. So the traffic flow data of A2 can be used to examine the performance of proposed algorithm when congestion.

The A4 motorway is part of the Rijksweg 4, which is another high-priority highway in the Netherlands, starting from Amsterdam to the Belgian border.

The A8 motorway starts from the A10 motorway at interchange Coenplein, ends at Zaandijk, less than 10 km.

There are some errors in the original data due to hardware failure, such as missing data or negative traffic flow values. The incorrect data was simply replaced or filled by averaging measurements at the same period in other weeks.

B. EVALUATION CRITERION

To measure the prediction performances of the proposed approach and other methods, two frequently used criteria, namely mean absolute percentage error (MAPE) and root mean square error (RMSE) are adopted. The RMSE measures the average differences between the predictions of an approach and the true values while the MAPE expresses the percentage of the differences. The smaller value of MAPE and RMSE, the better performance of the model. Two evaluation criteria are defined as followed:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2} \tag{25}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{\hat{y}_t - y_t}{y_t}\right| \times 100\% \tag{26}$$

which n is the number of the test samples, $\hat{y}_t$ denotes the predicted value and y_t denotes the true value of the tth value of data.

C. EXPERIMENT SETTING

All experiments are conducted at a workstation equipped with Intel® Core™ i7-4790U CPU@3.60GHz, 8G RAM. Our sample-rebalanced outlier-rejected k-nearest neighbor regression model was implemented with python 3.6 software environment.

The raw data in 1-min aggregation are reconstructed by 10-min average, since the goal of short-term traffic flow forecasting is not to predict minute by minute fluctuations. All the experiments use 60 minutes as the historical window, which constructed by 6 observed data points. The data set contains 5 weeks traffic flow data of each highway, corresponding to 1008 samples per week of each highway.

The raw data is 1-min aggregation and the datasets we used in our study are reconstructed by 10-min average traffic flow data. Because the goal of the study of short-term traffic flow forecasting should not be to predict minute by minute fluctuations. Therefore, the datasets are reconstructed by 10-min average traffic flow data. All the experiments use 60 minutes as the historical window, which constructed by 6 observed data points. The data set contains 5 weeks of traffic flow data of each highway, corresponding to 1008 samples per week of each highway.

The collected data are divided into two parts, while the data of the first four weeks are used as training set, and the rest are used as testing set. We use grid search technique to determine the optimal parameters, so the training data are divided into ten parts and nine of ten are used for training and the rest is for validation. The parameter k ranges from 3 to 50, and the candidate proportion of ω_1 and ω_2 is $1:1$, $2:3$, $1:2$, $2:5$, and $1:3$. After the grid search, the optimal parameter k, ω_1 and ω_2 are set to 15, 0.4 and 0.6, respectively.

The Euclidean distance is often used to measure the similarity between two data in traditional kNN algorithm, while DTW metric algorithm can automatically discover potential pattern in time series sequences. We also explore the potential to measure the similarity by DTW metric for traffic flow sequences. We model the kNN algorithm with the DTW metric, and evaluate it with the traffic flow data from A4. The forecasting performance of two different criteria on the A4 dataset is illustrated in Figure 3.

The experiment adopts two different evaluation criteria, e.g. RMSE and MAPE, to evaluate the forecasting performance of two different metrics. We divide the testing set into days and conduct the experiments. The upper figure shows MAPE of the predicted results for each day, while the lower figure shows RMSE. The results of the kNN algorithm based on Euclidean distance metric (Eu-kNN) are drawn with a green line, while the kNN algorithm based on DTW metric (DTW-kNN) are done with red line. As shown in Figure 3, DTW-kNN achieves high accuracy most of the time,

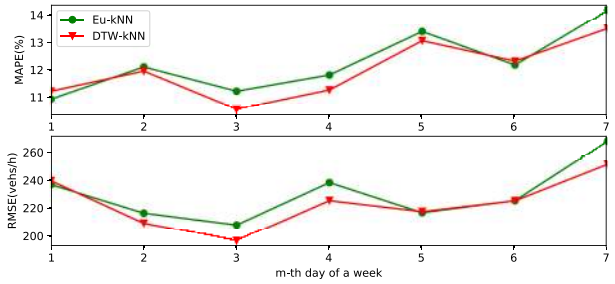

FIGURE 3. Forecasting performance with different metric.

which indicates that DTW metric is more suitable for short-term traffic flow forecasting than Euclidean distance metric.

In addition, we also conduct the following experience to evaluate how the number of neighbors k affect the performances of two different metrics. Evaluating the predictive performance of two different metric algorithms when the parameter k from 3 to 50. The forecasting performance of Eu-kNN and DTW-kNN on A4 dataset corresponding to Figure 4 and Figure 5. In Figure 4, The RMSE and MAPE decrease sharply until the number of neighbors k increases until 10 by Eu-kNN algorithm, after that the curves of RMSE and MAPE continuously rise.

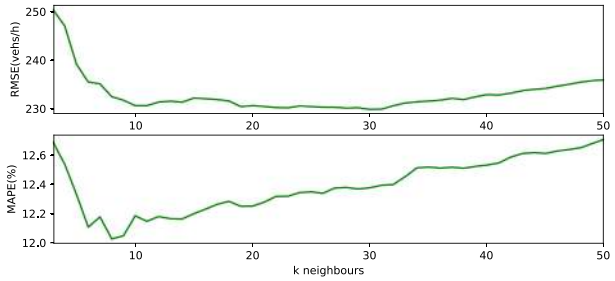

FIGURE 4. The performance of Eu-kNN with different number of neighbors k.

However, the RMSE and MAPE of DTW-kNN also decrease sharply before k is 20 in Figure 5 and stay steady after that point. This phenomenon indicates the DTW metric is more robust for traffic flow sequences. When the value of k becomes larger, the neighbor vectors obtained by DTW are more similar to the query vector than the nearest neighbor vectors obtained by Euclidean. Consequently, the RMSE and MAPE of DTW-kNN can be kept in a stationary state. From another perspective, it also reduces the number of noisy feature vectors.

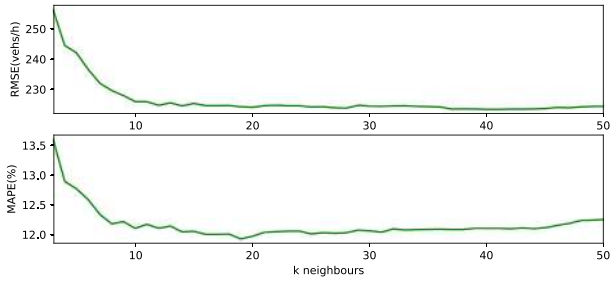

FIGURE 5. The performance of DTW-kNN with different number of neighbors k.

D. EXPERIMENT RESULTS

To evaluate the performance of our S^{r}O^rkNN^r model, we compare the forecasting results of with the following control models, which are the common models in intelligent transportation systems and often used as the baseline for a new method for short-term traffic flow forecasting, such as hybrid particle swarm optimization support vector regression method [18], autoregressive integrated moving average model [46]–[48], artificial neural network [22], Kalman Filtering method [49], decision tree [50], k-Nearest neighbors regression. Table 1 shows the forecasting result of each model on the each dataset of Amsterdam motorways.

TABLE 1. Forecasting performance comparison of different approaches on the datasets of Amsterdam motorways.

Model	Criteria	A1	A2	A4	A8
SVR	MAPE	14.34	12.22	12.23	12.48
	RMSE	329.09	259.74	253.66	190.30
ARIMA	MAPE	13.76	11.65	12.37	12.69
	RMSE	301.98	215.20	224.05	164.90
ANN	MAPE	12.61	10.89	12.49	12.53
	RMSE	299.64	212.95	225.86	166.50
KF	MAPE	12.46	10.72	12.62	12.63
	RMSE	332.03	239.87	250.51	187.48
DT	MAPE	12.08	10.86	12.34	13.62
	RMSE	316.57	224.79	243.19	238.35
LSTM	MAPE	12.82	11.06	13.71	12.56
	RMSE	294.52	211.31	224.68	168.91
kNN	MAPE	11.57	10.42	12.19	11.7
	RMSE	289.33	211.79	231.79	166.39
S^{r}O^rkNN^r	MAPE	11.27	10.00	11.60	11.63
	RMSE	281.3	203.54	218.37	162.38

1) SUPPORT VECTOR MACHINE REGRESSION (SVR)

In this study, we use SVR as a control experiment, and several important parameters need to be set beforehand. All parameters were set as suggested in Cai *et al.* [18] for the same datasets. The regression horizon is set the same as AR model and the radial basis function (RBF) is adopted to as the kernel function. The cost parameter C is set to the maximum difference between the traffic flow data. The width parameter γ for the RBF kernel is 3×10^{-6}, and the ϵ-insensitive loss for the SVR is 1.

2) AUTOREGRESSIVE INTEGRATED MOVING AVERAGE MODEL (ARIMA)

The ARIMA model is a frequently used method for time series related task, and widely used in traffic flow forecasting [46]. A typical ARIMA (p,d,q) model contains

three parameters, which p is the number of time lags of the autoregressive model, d is the number of times the data have had past values subtracted, q is the order of the moving average model. After examining the autocorrelation function (ACF) and the partial ACF, a set of alternative models are identified with different model orders. To determine the best model among the alternatives, diagnostics and Akaike information criteria (AIC) are used. Finally, the model orders $(7, 0, 3)$ with good diagnostics and good AIC value was chosen.

3) ARTIFICIAL NEURAL NETWORK (ANN)

The artificial neural network we used as control model was introduced by Zhu *et al.* [22], which is based on radial basis function neural network (RBFNN). The parameters of such network contain the number of hidden layers, the mean squared error goal, the spread of radial basis function, the maximum number of neurons in the hidden layer, and the number of neurons to add between displays. The parameters of such network are set as described in Table 2, where most of them are the same as [22] except *MN* for which we found a more suitable value.

TABLE 2. The configuration of the RBFNN model.

Parameters	Value
Hidden layers	1
Goal	0.001
Spread	2000
MN	40
DP	Default

4) KALMAN FILTERING METHOD (KF)

Kalman filtering is an algorithm for optimal estimation of system state by observation data of the system. The prediction result of the Kalman filtering method are easily affected by noisy data. A wavelet denoising procedure proposed by Xie *et al.* [49] is employed to preprocess the traffic flow data and improve the performance of Kalman filtering. We use Daubechies 4 as the mother wavelet as suggesting in Xie *et al.* [49].

5) DECISION TREES (DT)

The decision tree model we used in this study is based on the classification and regression tree (CART) to forecast traffic flows. The CART does not need any prior hypothesis and has strong robustness. The CART is detailed in Xu *et al.* [50].

6) LONG SHORT-TERM MEMORY (LSTM)

LSTM network [51] is a special kind of recurrent neural networks (RNN), which is developed to capture time dependence in a long period. The LSTM network parameters contain the number of hyperparameter units, the batch size, the epochs and the validation split. The model parameters are set as described in Table 3, which are optimized by grid search.

TABLE 3. The parameters of the LSTM model.

Parameters	Value
Hyperparameter units	256
Batch size	32
Epochs	50
Validation split	0.05

7) *k*-NEAREST NEIGHBOR (*k*NN)

The algorithm predicts the properties of a query vector by using the properties of several vectors closest to the query vector.

Comparing with the prediction results of the control models mentioned above. As shown in Table 1, $\mathrm{S}^r\mathrm{O}^r k\mathrm{NN}^r$ achieves the best performance. For example, comparing with the basic kNNa, which achieves the highest accuracy in all control experiments, the RMSEs of $\mathrm{S}^r\mathrm{O}^r k\mathrm{NN}^r$ decrease by 2.78%, 3.9%, 5.79%, and 3.43% at A1, A2, A4 and A8, respectively.

Besides, we visualize the deviation between the forecasting results of the $\mathrm{S}^r\mathrm{O}^r k\mathrm{NN}^r$ and the groundtruth in Figure 6. The red line represents the prediction, and the green line represents the groundtruth. As shown in Figure 6, most of the time, the prediction fit the groundtruth well. The related error, e.g. the difference between the prediction and the groundtruth divided by the groundtruth, drawn in the blue line. In some cases, the predicting results have a large related error during peak period or low flow period. This phenomenon is due to the severe changes in traffic flow data on the one hand, and it also caused by inherent defects, which often occurs in the low flow period early in the morning or late at night. At these moments, the traffic flow usually operate in free speed, so the large related error is insignificant.

Figure 7 shows the detailed prediction results of various methods for nine typical but challenge scenarios. The $\mathrm{S}^r\mathrm{O}^r k\mathrm{NN}^r$ model achieves more accurate results than other models. Due to the outlier rejection and sample rebalancing mechanism, the model achieve better performance of traffic flow forecasting.

To evaluate the $\mathrm{S}^r\mathrm{O}^r k\mathrm{NN}^r$ model more comprehensively, we calculate the coefficient of determination of the $\mathrm{S}^r\mathrm{O}^r k\mathrm{NN}^r$ model on the dataset of A1, A2, A4 and A8. The results are shown in Table 4. The coefficient of determination, termed R^2, is the proportion of the variance in the dependent variable that is predictable from the independent variables. The value of R^2 ranges from 0 to 1. A higher coefficient is an indicator ofbetter goodness of fit for the observations. The R^2 values of our $\mathrm{S}^r\mathrm{O}^r k\mathrm{NN}^r$ on four datasets all exceed 0.95, which indicates all movements of dependent variable are largely explained by the independent variables we are interested in. The results show the superior performances of our $\mathrm{S}^r\mathrm{O}^r k\mathrm{NN}^r$ model.

TABLE 4. The coefficient of determination (R^2) for the four datasets.

Model	A1	A2	A4	A8
$\mathrm{S}^r\mathrm{O}^r k\mathrm{NN}^r$	0.95	0.98	0.96	0.95

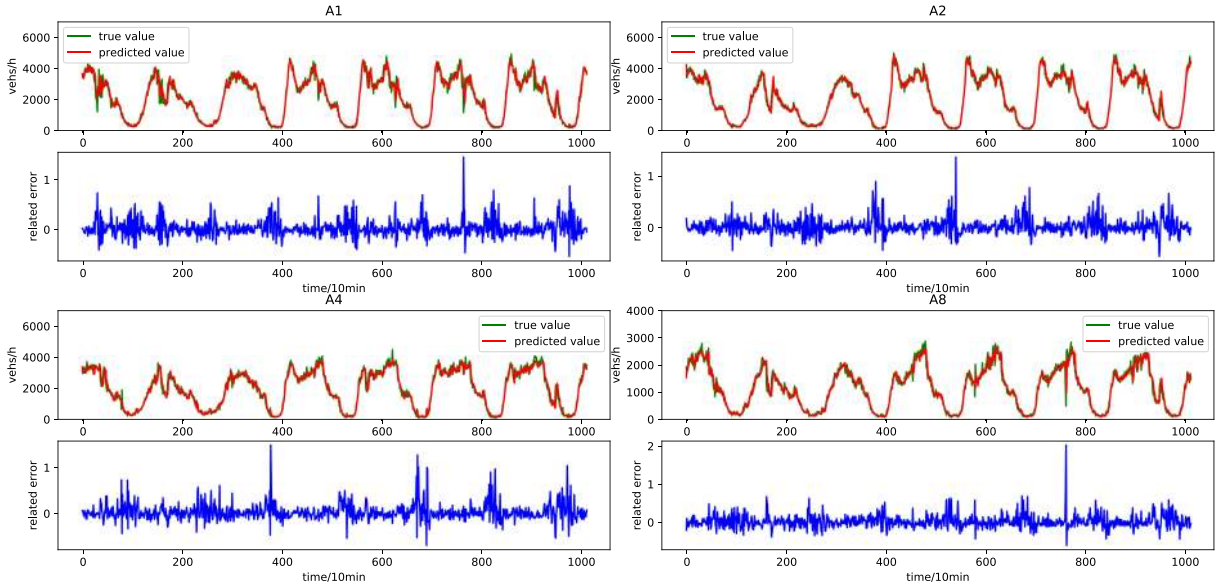

FIGURE 6. The predictions of proposal and the true value in a week, and the prediction related error.

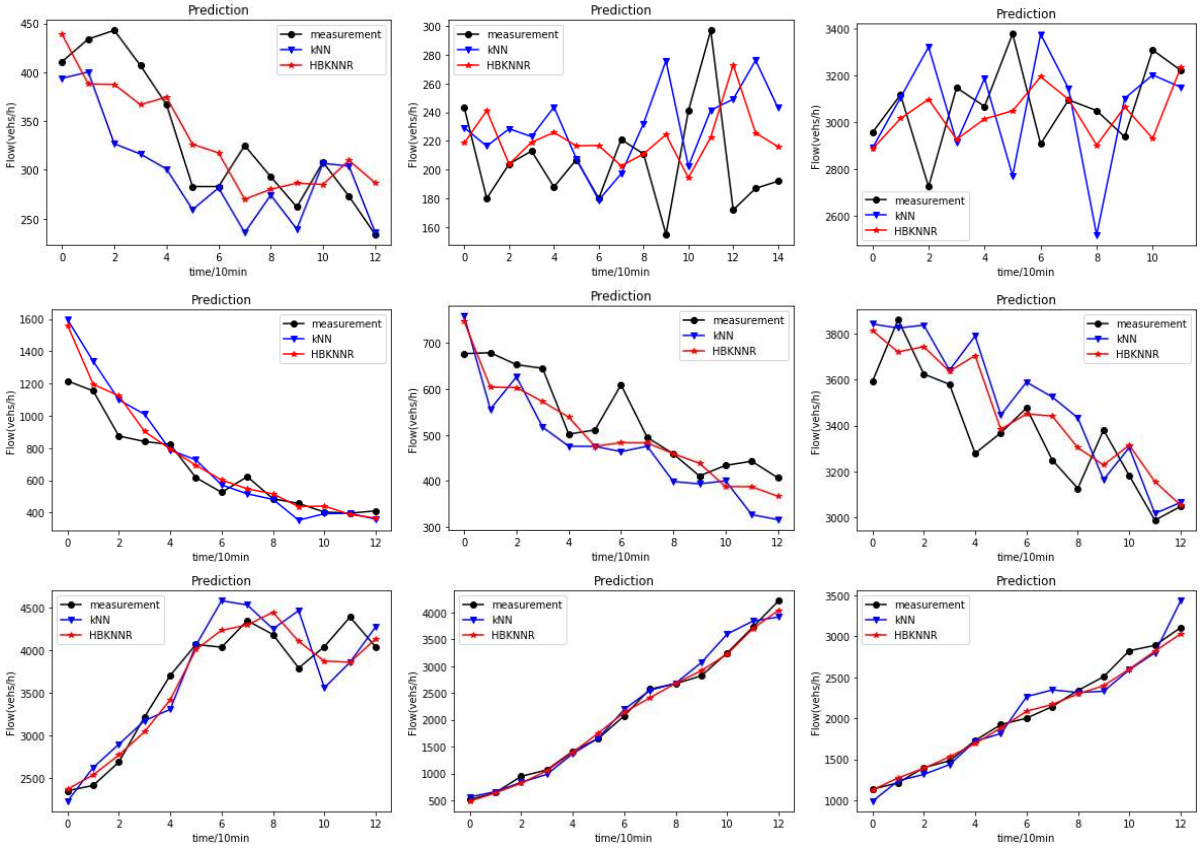

FIGURE 7. The detailed prediction results of various methods in nine typical scenarios.

The model is also applicable to time series forecasting problems. We also plan to further improve our model by considering the concept change n the traffic flow patterns, and further evaluate our model on other forecasting problems. Furthermore, the proposed model can be easily extended to other temporal related applications, such as

multi-dimensional time series prediction [52], [53], prediction of stable bitstream [54], [55], recommendation systems [56] or temporal human emotion computing [57], etc.

V. CONCLUSION

In this paper, we present a sample-rebalanced outlier-rejected *k*-nearest neighbour regression model for short-term traffic flow forecasting. Our model reconstructs balanced training sets to address the imbalance issue of the traffic flow data, and considers both local and global patterns of the traffic flow data. Furthermore, we introduce a new similarity metric for the evolutional traffic flow dynamic. The experiments on four benchmark datasets comparing with state-of-the-art models show the presented model can significantly improve the accuracy of short-term traffic flow forecasting.

REFERENCES

[1] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 50–64, Jun. 2014.

[2] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Sep. 2014.

[3] M. Maaref and Z. M. Kassas, "A closed-loop map-matching approach for ground vehicle navigation in gnss-denied environments using signals of opportunity," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[4] X. Kong, M. Li, T. Tang, K. Tian, L. Moreira-Matias, and F. Xia, "Shared subway shuttle bus route planning based on transport data analytics," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1507–1520, Oct. 2018.

[5] X. Li, "The symmetric intersection design and traffic control optimization," *Transp. Res. C, Emerg. Technol.*, vol. 92, pp. 176–190, Jul. 2018.

[6] Z. Zheng and D. Su, "Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm," *Transp. Res. C, Emerging Technol.*, vol. 43, pp. 143–157, Jun. 2014.

[7] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 871–882, Jun. 2013.

[8] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, "Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 644–654, Jun. 2012.

[9] H.-F. Yang, T. S. Dillon, E. Chang, and Y.-P. P. Chen, "Optimized configuration of exponential smoothing and extreme learning machine for traffic flow forecasting," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 23–34, Jan. 2019.

[10] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003.

[11] H. Zhang, X. Wang, J. Cao, M. Tang, and Y. Guo, "A multivariate short-term traffic flow forecasting method based on wavelet analysis and seasonal time series," *Appl. Intell.*, vol. 48, no. 10, pp. 3827–3838, Oct. 2018.

[12] T. Zhou, D. Jiang, Z. Lin, G. Han, X. Xu, and J. Qin, "Hybrid dual Kalman filtering model for short-term traffic flow forecasting," *IET Intell. Transp. Syst.*, vol. 13, no. 6, pp. 1023–1032, Jun. 2019.

[13] L. Cai, Z. Zhang, J. Yang, Y. Yu, T. Zhou, and J. Qin, "A noise-immune Kalman filter for short-term traffic flow forecasting," *Phys. A, Stat. Mech. Appl.*, vol. 536, Dec. 2019, Art. no. 122601.

[14] S. Zhang, Y. Song, D. Jiang, T. Zhou, and J. Qin, "Noise-identified Kalman filter for short-term traffic flow forecasting," in *Proc. 15th Int. Conf. Mobile Ad-Hoc Sensor Netw. (MSN)*, 2019, pp. 1–5.

[15] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3634–3640.

[16] X. Luo, D. Li, Y. Yang, and S. Zhang, "Spatiotemporal traffic flow prediction with KNN and LSTM," *J. Adv. Transp.*, vol. 2019, pp. 1–10, Feb. 2019.

[17] S. Liang, M. Ma, S. He, and H. Zhang, "Short-term passenger flow prediction in urban public transport: Kalman filtering combined k-nearest neighbor approach," *IEEE Access*, vol. 7, pp. 120937–120949, 2019.

[18] L. Cai, Q. Chen, W. Cai, X. Xu, T. Zhou, and J. Qin, "SVRGSA: A hybrid learning based model for short-term traffic flow forecasting," *IET Intell. Transp. Syst.*, vol. 13, no. 9, pp. 1348–1355, Sep. 2019.

[19] W. Cai, D. Yu, Z. Wu, X. Du, and T. Zhou, "A hybrid ensemble learning framework for basketball outcomes prediction," *Phys. A, Stat. Mech. Appl.*, vol. 528, Aug. 2019, Art. no. 121461.

[20] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive multi-kernel SVM with spatial–temporal correlation for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2001–2013, Jun. 2019.

[21] W. Cai, J. Yang, Y. Yu, Y. Song, T. Zhou, and J. Qin, "PSO-ELM: A hybrid learning model for short-term traffic flow forecasting," *IEEE Access*, vol. 8, pp. 6505–6514, 2020.

[22] J. Z. Zhu, J. X. Cao, and Y. Zhu, "Traffic volume forecasting based on radial basis function neural network with the consideration of traffic flows at the adjacent intersections," *Transp. Res. C, Emerg. Technol.*, vol. 47, pp. 139–154, Oct. 2014.

[23] T. Pamula, "Impact of data loss for prediction of traffic flow on an urban road using neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 1000–1009, Mar. 2019.

[24] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[25] T. Zhou, G. Han, X. Xu, Z. Lin, C. Han, Y. Huang, and J. Qin, "δ-agree Adaboost stacked autoencoder for short-term traffic flow forecasting," *Neurocomputing*, vol. 247, pp. 31–38, 2017.

[26] T. Zhou, G. Han, X. Xu, C. Han, Y. Huang, and J. Qin, "A learning-based multimodel integrated framework for dynamic traffic flow forecasting," *Neural Process. Lett.*, vol. 49, no. 1, pp. 407–430, Feb. 2019.

[27] Y. Zhang, Y. Zhang, and A. Haghani, "A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 65–78, Jun. 2014.

[28] Z. Yu, H. Chen, J. Liu, J. You, H. Leung, and G. Han, "Hybrid k-nearest neighbor classifier," *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1263–1275, Jun. 2015.

[29] S. Li, Z. Shen, and G. Xiong, "A k-nearest neighbor locally weighted regression method for short-term traffic flow forecasting," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 1596–1601.

[30] H. Hong, W. Huang, X. Xing, X. Zhou, H. Lu, K. Bian, and K. Xie, "Hybrid multi-metric k-nearest neighbor regression for traffic flow prediction," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 2262–2267.

[31] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 62, pp. 21–34, Jan. 2016.

[32] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognit.*, vol. 74, pp. 406–421, Feb. 2018.

[33] L. Blouvshtein and D. Cohen-Or, "Outlier detection for robust multi-dimensional scaling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2273–2279, Sep. 2019.

[34] D. Chakraborty, V. Narayanan, and A. Ghosh, "Integration of deep feature extraction and ensemble learning for outlier detection," *Pattern Recognit.*, vol. 89, pp. 161–171, May 2019.

[35] J.-T. Jeng, C.-M. Chen, S.-C. Chang, and C.-C. Chuang, "IPFCM clustering algorithm under Euclidean and Hausdorff distance measure for symbolic interval data," *Int. J. Fuzzy Syst.*, vol. 21, no. 7, pp. 2102–2119, Oct. 2019.

[36] R. Zhang, F. Nie, M. Guo, X. Wei, and X. Li, "Joint learning of fuzzy k-means and nonnegative spectral clustering with side information," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2152–2162, May 2019.

[37] G. Wang, Y. Chen, and X. Zheng, "Gaussian field consensus: A robust nonparametric matching method for outlier rejection," *Pattern Recognit.*, vol. 74, pp. 305–316, Feb. 2018.

[38] Y. Song, Z. Yu, T. Zhou, J. Y.-C. Teoh, B. Lei, C. Kup-Sze, and J. Qin, "CNN in CT image segmentation: Beyond loss function for exploiting ground truth images," in *Proc. IEEE Int. Symp. Biomed. Imag. (ISBI)*, 2020, pp. 1–4.

[39] X. Tang, Y. Dai, Q. Liu, X. Dang, and J. Xu, "Application of bidirectional recurrent neural network combined with deep belief network in short-term load forecasting," *IEEE Access*, vol. 7, pp. 160660–160670, 2019.

[40] D. Li, Y. Zhao, and Y. Li, "Time-series representation and clustering approaches for sharing bike usage mining," *IEEE Access*, vol. 7, pp. 177856–177863, 2019.
[41] Y. Chen, F. Guo, and J. Ruan, "Constructing odd-variable RSBFs with optimal algebraic immunity, good nonlinearity and good behavior against fast algebraic attacks," *Discrete Appl. Math.*, vol. 262, pp. 1–12, Jun. 2019.
[42] Y. Chen, F. Guo, H. Xiang, W. Cai, and X. He, "Balanced odd-variable RSBFs with optimum AI, high nonlinearity and good behavior against FAAs," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E102.A, no. 6, pp. 818–824, Jun. 2019.
[43] Z. Yu, Z. Niu, W. Tang, and Q. Wu, "Deep learning for daily peak load forecasting—A novel gated recurrent neural network combining dynamic time warping," *IEEE Access*, vol. 7, pp. 17184–17194, 2019.
[44] P. Vincent and Y. Bengio, "K-local hyperplane and convex distance nearest neighbor algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 985–992.
[45] Y. Wang, J. H. Van Schuppen, and J. Vrancken, "Prediction of traffic flow at the boundary of a motorway network," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 214–227, Feb. 2014.
[46] H. Moayedi and M. Masnadi-Shirazi, "Arima model for network traffic prediction and anomaly detection," in *Proc. Int. Symp. Inf. Technol.*, vol. 4, Sep. 2008, pp. 1–6.
[47] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2664–2675, Mar. 2011.
[48] G. Comert and A. Bezuglov, "An online change-point-based model for traffic parameter prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1360–1369, Sep. 2013.
[49] Y. Xie, Y. Zhang, and Z. Ye, "Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 22, no. 5, pp. 326–334, 2007.
[50] Y. Xu, Q.-J. Kong, and Y. Liu, "Short-term traffic volume prediction using classification and regression trees," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2013, pp. 493–498.
[51] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.
[52] T. Zhou, G. Han, B. N. Li, Z. Lin, E. J. Ciaccio, P. H. Green, and J. Qin, "Quantitative analysis of patients with celiac disease by video capsule endoscopy: A deep learning method," *Comput. Biol. Med.*, vol. 85, pp. 1–6, Jun. 2017.
[53] B. N. N. Li, X. Wang, R. Wang, T. Zhou, R. Gao, E. J. Ciaccio, and P. H. Green, "Celiac disease detection from videocapsule endoscopy images using strip principal component analysis," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, to be published.
[54] Y. Chen, F. Guo, Z. Gong, and W. Cai, "One note about the Tu-Deng conjecture in case $w(t) = 5$," *IEEE Access*, vol. 7, pp. 13799–13802, 2019.
[55] Y. Chen, L. Zhang, D. Tang, and W. Cai, "Translation equivalence of Boolean functions expressed by primitive element," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E102.A, no. 4, pp. 672–675, Apr. 2019.
[56] D. Jiang, Z. Liu, L. Zheng, and J. Chen, "Factorization meets neural networks: A scalable and efficient recommender for solving the new user problem," *IEEE Access*, to be published.
[57] D. Jiang, K. Wu, D. Chen, G. Tu, T. Zhou, A. Garg, and L. Gao, "A probability and integrated learning based classification algorithm for high-level human emotion recognition problems," *Measurement*, vol. 150, Jan. 2020, Art. no. 107049.

LINGRU CAI received the Ph.D. degree from the Huazhong University of Science and Technology, in 2010. She is currently an Associate Professor with the Department of Computer Science, Shantou University, China. Her research interests include intelligent transportation systems, modeling and simulation of complex systems, game theory, and machine learning.

YIDAN YU is currently pursuing the master's degree with the Department of Computer Science, College of Engineering, Shantou University, China. Her research interests include urban computing and machine learning.

SHUANGYI ZHANG is currently pursuing the master's degree with the Department of Computer Science, College of Engineering, Shantou University, China. Her research interests include urban computing and machine learning.

YOUYI SONG is currently pursuing the Ph.D. degree with the Centre for Smart Health, School of Nursing, The Hong Kong Polytechnic University. His research interests include urban computing and machine learning.

ZHI XIONG received the Ph.D. degree from Wuhan University. He is currently an Associate Professor with the Department of Computer Science, Shantou University. His research interests include intelligent transportation systems, urban computing, and machine learning.

TENG ZHOU is currently an Assistant Professor with the Department of Computer Science, Shantou University. He also serves as a Research Associate with the Center of Smart Health, The Hong Kong Polytechnic University. His research interests include intelligent transportation systems and machine learning.