Knowledge-Based Systems 163 (2019) 472–484

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Forecasting short-term traffic speed based on multiple attributes of adjacent roads

Dongjin Yu [a,*], Chengfei Liu [b], Yiyu Wu [a], Sai Liao [a], Tarique Anwar [b], Wanqing Li [a], Chengbiao Zhou [c]

[a] College of Computer, Hangzhou Dianzi University, Hangzhou, China
[b] Swinburne University of Technology, Melbourne, Australia
[c] Hangzhou Trustway Technology Company Limited, China

HIGHLIGHTS

- We consider only traffic attributes most relevant to speed and also their weights.
- We consider the differences of impacts on speed among adjacent roads.
- We employ piecewise correlation function between traffic speed and attributes.
- We adopt Jenks clustering with dynamic programming to determine segment intervals.
- We validate our approach based on the real data collected from two large cities.

ARTICLE INFO

Article history:
Received 9 May 2018
Received in revised form 28 August 2018
Accepted 5 September 2018
Available online 15 September 2018

Keywords:
Traffic speeds
Short-term
Speed prediction
Adjacent roads
Traffic attributes

ABSTRACT

Forecasting the short-term speed of moving vehicles on roads plays a vital role on traffic control and trip planning, which however still remains a challenging task when the high accuracy is required. In this paper, we propose a novel approach to the short-term traffic speed forecasting, which takes into account the influence of different traffic attributes, such as traffic flow, traffic speed, road occupancy and traffic density, of adjacent roads on the traffic speed. In addition, in order to obtain the more accurate relation between traffic speed and traffic attributes, we employ the idea of piecewise correlation function and adopt the Jenks clustering method with dynamic programming to determine the segment intervals of relation. We validate our approach based on the real data collected from Wenzhou and Hangzhou, two large cities located in eastern China. The extensive experimental results show that, compared with the state-of-the-art approaches, our approach has the higher stability and accuracy, especially for 5-minute and 10-minute speed prediction.

1. Introduction

The accurate estimation of travel condition is of great significance to ease traffic congestion [1], which not only reduces travel time, but also saves energy and reduces air pollution. As the speed of vehicles is one of the most important parameters of traffic attributes, the short-term prediction of traffic speed is extremely important for route planning and traffic regulations [2]. Here, the so-called short-term traffic speed forecasting refers to the effective use of real-time traffic data to predict the changes of vehicle speeds on a certain road or road segment within a period of short time in near future, for example, 5 min. However, the research on short-term forecasting of traffic speed still remains to be immature although numerous works have been done during the past decade. To the best of our knowledge, most of the current approaches predict the future traffic speeds based on the previous speeds. In fact, however, the future traffic speed is closely related to many recent traffic attributes, such as traffic flow, road occupancy and traffic density, but not just only traffic speed, in its adjacent area.

In this paper, we propose an approach to the traffic speed prediction based on those related traffic attributes of adjacent roads. Generally speaking, the adjacent roads are those related to the target road, but are not necessarily directly connected to the target road. The change in traffic attributes of an adjacent road will affect the traffic speed of the target road. However, choosing too many adjacent roads will certainly increase the complexity of the forecasting algorithm and thus decrease its running performance, while choosing too few ones will degrade the forecasting accuracy. To address this problem, in our approach, we firstly determine the K most relevant adjacent roads for each road, and then select

* Corresponding author.
E-mail address: yudj@hdu.edu.cn (D. Yu).

https://doi.org/10.1016/j.knosys.2018.09.003
0950-7051/

M traffic attributes most relevant to the traffic speed and calculate their corresponding weights of contribution based on the gray correlation analysis. Here, the reason we introduce the gray correlation analysis method lies in that it is applicable to various sample sizes with uncomplicated computation no matter whether the samples are regular or not. Secondly, due to the intricate relationship between traffic attributes and traffic speed, it is difficult to use a single function to represent the transition between the traffic attribute and the traffic speed. So we introduce the idea of piecewise function to obtain the impact function for each traffic attribute under each time slot on traffic speed. In particular, we use Jenks clustering based on dynamic programming to determine the segmentation interval for each traffic attribute. Jenks clustering has the advantages of being simple, efficient, and stable, compared with the commonly used k-means clustering. Finally, we combine the adjacent roads and related traffic attributes for the short-time traffic speed forecasting, and obtain the relevant parameters through the particle swarm optimization (PSO) algorithm.

The main contributions of this paper are as follows. (1) Different from some state-of-the-art methods which obtain the traffic attributes related to the traffic speed by using all recent traffic data, we obtain only those traffic attributes most relevant to the traffic speed and also their corresponding weights. Since we consider the differences of impacts among roads, our approach achieves higher accuracy. (2) We employ Jenks clustering based on dynamic programming to cluster each traffic attribute, so as to determine the number of segments and the segment interval. (3) We validate our approach based on large real data. The extensive results demonstrate the effectiveness and efficiency of our approach.

The rest of the paper is organized as follows. Section 2 presents the related work, and Section 3 describes some definitions, the problem statement, and the overall architecture of the approach to the short-term traffic speed approach. Afterwards, Section 4 proposes the approach to the short-term traffic speed forecasting in detail, whereas Section 5 presents and analyzes the experimental results based on real data. Finally, we conclude the paper, and outline the future work in Section 6.

2. Related work

Accurate prediction of traffic speed is of great help to trip planning and traffic regulation. During the past decade, numerous works have been done on the prediction of traffic speeds. Researchers have proposed many prediction approaches mainly based on statistics, neural networks, big data, machine learning, etc. Also many researchers integrate several approaches to obtain a new one.

Speed prediction heavily relies on the historical traffic data. Therefore, many approaches extensively employ the technology of data mining and statistics. For example in [3], Xu et al. introduced a short-term traffic flow forecasting method which took into account the correlation of related roads in the same area. In [4], Myung et al. proposed a KNN prediction model based on data from both the vehicle detector system and the automatic toll collection system, which minimized the limitations of each set to enhance the prediction accuracy. Besides, Clark S presented a model based on multivariate non-parametric regression, whose accuracy was demonstrated when predicting the traveling time of London railways [5], whereas Bing Zhang et al. proposed a speed clustering approach that incorporated multi-lane information with partitioning and merging [6]. On the other hand, the emergence of big data technology in the recent years contributes greatly to the traffic speed forecast. For example in [7], Chen et al. introduced a big-data approach, which combined KNN and Gaussian process regression based on the MapReduce architecture for efficient and robust traffic speed prediction. In addition, in [8], Jeon et al. proposed a traffic velocity prediction approach. They constructed a large data processing framework which was capable to handle the overall prediction process such as Monte Carlo simulations based on massive historical data.

Deep learning and artificial neural networks have demonstrated their effectiveness and wide application in traffic speed prediction. Hinsbergen, as one of pioneers, applied Bayesian neural networks for the prediction of stochastic travel times in urban networks, and proved that the neural networks were capable of predicting the so-called ‘low-frequency trend’ [9]. From then on, especially in recent years, many novel works based on LSTM and CNN have been proposed. For example, Ma et al. presented a Long Short-Term Memory (LSTM) neural network to predict travel speed using microwave detector data [10], while Jiang et al. proposed a two-level vehicle speed prediction system for highways based on Neural Networks and Hidden Markov models [11]. In [12], Ma et al. proposed a Convolutional Neural Network (CNN)-based method for large-scale traffic speed prediction that learned traffic as images and predicts large-scale, network-wide traffic speed with a high accuracy. Meanwhile, Wang et al. proposed a deep learning method with a structure of Error-feedback Recurrent Convolutional Neural Network (eRCNN) for continuous traffic speed prediction. By integrating the spatio-temporal traffic speeds of contiguous road segments as an input matrix, eRCNN explicitly leveraged the implicit correlations among nearby segments to improve the predictive accuracy [13]. Recently, some researchers employed emerging deep learning techniques for short term speed prediction and achieved rather good results [14,15].

Speed pattern analysis is frequently employed to predict traffic speed. In [16], Kehagias et al. proposed a non-parametric short-term speed pattern recognition technique based on speed dynamics. They suggested lowering the dimension of the available dataset for enabling more efficient deployment of clustering techniques in terms of computational resource efficiency and forecasting accuracy. In addition, Asif et al. proposed the unsupervised learning methods, such as k-means clustering, principal component analysis and self-organizing mapping, to exploit the spatial–temporal patterns to predict large-scale traffic speeds [17]. It is also worth mentioning that in [18], Fabrizi et al. presented a pattern matching approach for traffic speed forecasting based on an operating Floating-Car Data (FCD) system. The accuracy was demonstrated when predicting the traffic speed of the Rome Ring Road. Another work, led by Xia et al., employed a seasonal ARIMA model and quantified the seasonal recurrence pattern of traffic conditions [19].

Traffic speed prediction can improve the accuracy of travel time estimates. For this purpose, Jenelius et al. presented a statistical model for urban road network travel time estimation using vehicle trajectories obtained from low frequency GPS probes as observations [20]. In addition, in [21], a three-layer neural network model based on the information collected by probe vehicles was proposed to estimate the complete travel time for individual probe vehicle traversing the link. In [22], an algorithm was proposed for predicting the remaining travel times of long-range trips, which was applied to logistics planning. However, it made use of nonparametric distribution-free regression models, which was applicable only in the presence of a sufficiently large database.

Short-term traffic speed forecast, compared with long-term speed forecast, pays more attention to its precision and traffic dynamics. In [23], Rasyidi et al. proposed an improved k-Nearest-Neighbor algorithm to predict short-term traffic speed, and its effect was proven to be superior to both linear regression and model tree. Considering the relationship between road sections, H. Hu et al. combined the KNN method with the seed section to predict the sections with sparse data for short-term traffic speed forecast [24]. Meanwhile, considering the awareness of the time of the day, P. Dell A et al. proposed a multivariate nearest neighbor regression (TaM-NNR) algorithm to predict the link speed in short-term [25]. Besides, W. Hu et al. presented a Hybrid PSO-SVR method to forecast short-term traffic flow, which employed particle swarm optimization (PSO) to search optimal SVR parameters [26], whereas Yao et al. proposed a support vector machine model composed of spatial and temporal parameters to predict short-term traffic speed [27]. Furthermore, in [28], Wang et al. proposed a short-term traffic forecasting hybrid model based on Chaos-Wavelet Analysis-Support Vector Machine model (referred to as C-WSVM model). On the other hand, it is difficult to predict the traffic speed in real time, which is however a must for short-term forecasting. To address this problem, Anil et al. presented a comprehensive solution for implementing a processing module on traffic cameras that is capable of tracking every vehicle in the camera frame and estimating its speed in real-time [29]. Some other real-time estimation approaches include using data-filtering procedure based on a varying data validity window within a sampling interval [30], and using infrastructure-free vehicular networks and macroscopic traffic flow model [31]. In [32], an aggregation approach was proposed for traffic flow prediction that is based on the moving average (MA), exponential smoothing (ES), autoregressive MA (ARIMA), and neural network (NN) models.

Predicting traffic speeds always requires exogenous data such as GPS, loop detectors and mobile phones. For examples, Bin Deng et al. proposed a method for estimating actual, congested link speeds from taxi GPS traces with low-sampling frequency. The method was based on a path inference process and applied over a detailed road network in a large city region [33]. To improve the accuracy of speed estimation, Zhang et al. proposed a travel-time-based method to aggregate the estimation results of the cellular probe system and loop detectors [34]. Different from above mentioned ones, with the temporal sequences of vehicle count, Katsuki et al. proposed a velocity estimation approach that did not require tracking any vehicles or using any labeled data [35].

To obtain real-time, accurate traffic flow information is the key to accurately forecasting short-term traffic speed. However, it is rather difficult to obtain the traffic information of the whole road network based on the fixed detection equipments. On the other hand, the GPS data captured by mobile detection equipments have the advantages of short construction time, low maintenance cost, high accuracy and reliability, and satisfactory real-time performance. Besides, the speeds in short-term are easily influenced by many factors such as traffic flow and road occupancy in addition to the previous traffic speeds. Different from the state-of-the-art ones, the approach proposed in this paper takes into account the influence of different traffic attributes, such as traffic flow, traffic speed, road occupancy and traffic density, of adjacent roads. The experiment shows that such approach improves the prediction accuracy by integrating multiple traffic attributes.

3. Problem statement and overall architecture

3.1. Preliminaries and problem statement

Definition 1 (*Road Network*). A road network is composed of a series of vertices (or endpoints of roads) and directed edges (or roads), denoted *as* $\boldsymbol{RN} = (\boldsymbol{I}, \boldsymbol{R})$, in which $\boldsymbol{I} = \{l_1, l_2, \ldots, l_{nl}\}$ represents a set of nl vertices and $\boldsymbol{R} = \{r_1, r_2, \ldots, r_{nr}\}$ a set of nr directed edges. Here, the edge (or road), connecting two vertices, can be represented by a 5-dimensional tuple: $\langle rid, len, pstart, pmiddle, pend\rangle$, in which $rid, len, pstart, pmiddle$ and $pend$ represent the identifier of the edge (or road), its length, and its start, middle and end vertices (or endpoints of road) respectively.

Definition 2 (*Trajectory*). The trajectory of a moving vehicle, recorded by GPS, is defined as a series of geo-location points with timestamps, or ***TP-Set*** $= \{p_1, p_2, \ldots, p_{np}\}$, where each point $p_i = (t, lon, lat)$ includes its timestamp t, longitude lon and latitude lat. The set of trajectory points of all floating vehicles during a continuous period of time from sT to eT is denoted as ***TP-Set***(sT, eT).

Definition 3 (*Sample Data Set of Traffic Attributes*). It is assumed that there are M categories of traffic attributes $s_1, s_2, \ldots, s_M$, such as traffic flow, traffic speed, road occupancy and traffic density. Given p different time points, the sample dataset of traffic attributes is denoted as $\boldsymbol{TC} = \{\boldsymbol{TC}_1, \boldsymbol{TC}_2, \ldots, \boldsymbol{TC}_i, \ldots, \boldsymbol{TC}_p\}$, in which $\boldsymbol{TC}_i = (\boldsymbol{tc}_{i,1}, \boldsymbol{tc}_{i,2}, \ldots, \boldsymbol{tc}_{i,j}, \ldots, \boldsymbol{tc}_{i,nr})$ represents the traffic attributes of all nr roads at time point i, in which $\mathbf{tc}_{i,j} = (ts^1_{i,j}, ts^2_{i,j}, \ldots, ts^M_{i,j}, \ldots, ts^T_{i,j})$ gives the values of all T traffic attributes of road r_j at time point i. Here, the sampling period is divided into different time points (or time slots), with each having equal length of Δt minutes.

Definition 4 (*Sample Data Set of Traffic Speed*). Given p different time points, the sample dataset of traffic speed is denoted as $\boldsymbol{SP} = \{\boldsymbol{SP_1}, \boldsymbol{SP_2}, \ldots, \boldsymbol{SP_i}, \ldots, \boldsymbol{SP_p}\}$, where $\boldsymbol{SP}_i = (sp_{i,1}, sp_{i,2}, \ldots, sp_{i,j}, \ldots, sp_{i,nr})$ represents the traffic speeds of all nr roads at time point i. It should be noted that the traffic speed is also a kind of traffic attribute.

Problem Statement. *For a road network* $\boldsymbol{RN} = (\boldsymbol{I}, \boldsymbol{R})$, *given the traffic attributes (including traffic speed) of each road for p time points before time point t, i.e.,* $\boldsymbol{TC}$, *the problem considered in this paper is to forecast the traffic speed of each road in the immediate future time point* $t + h$, *i.e.,* $\boldsymbol{SP}_{t+h}$.

3.2. Overall architecture

In order to precisely forecast the traffic speed in near future with the acceptable computational complexity, we propose a forecasting approach, called NN_Forecast, based on the road network and the past traffic data of moving vehicles. NN_Forecast considers the impact of traffic attributes of the adjacent roads, such as traffic flow, traffic speed, road occupancy and traffic density, on the traffic speed in near future. Of course, NN_Forecast requires the data of relevant traffic attributes, usually obtained from moving vehicles, and also the road network, when initiating its model. Fig. 1 shows its architecture, which contains the following four steps.

(1) Finding adjacent roads. This step is to use floating vehicle trajectory data to discover K most relevant adjacent roads for each target road, whose changes of traffic attributes will affect traffic speed of target road. It is noteworthy that the adjacent roads of one road are not necessarily geographically connected. Here, we need to find the most appropriate K value, because choosing too many roads will increase the complexity of the forecasting algorithm and thus decrease its running performance, and choosing too few roads reduces the prediction accuracy due to the limited data.
(2) Finding relevant traffic attributes. As we have found the K most relevant adjacent roads for each road from the previous step, now we need to find M traffic attributes most relevant to the future traffic speed for each road from its K most relevant adjacent roads. Here, we adopt the gray correlation analysis method to find the relation between the traffic speed of the road and the traffic attributes of its K most relevant adjacent roads, and obtain the M most relevant traffic attributes and their degrees of correlation.
(3) Clustering traffic attributes. In order to find the precise correlation function between each relevant traffic attribute of each adjacent road and the traffic speed, we employ the Jenks clustering algorithm to divide the traffic attribute

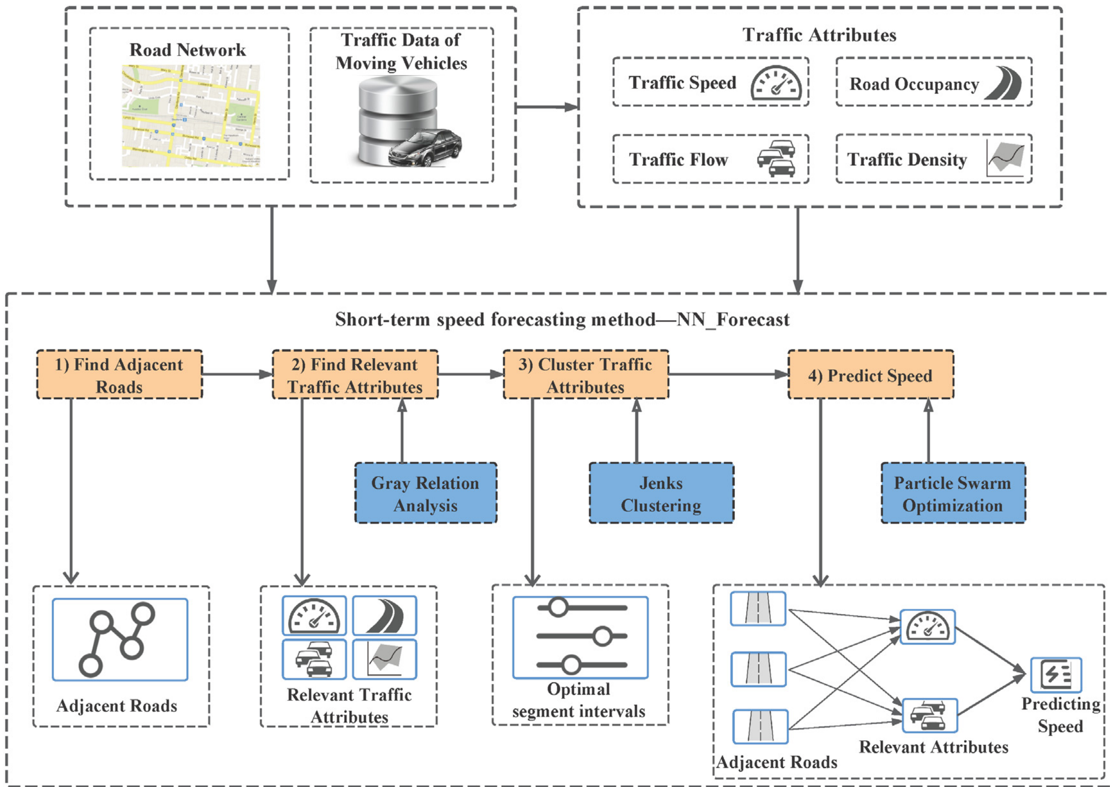


Fig. 1. The overall architecture of speed prediction approach.

values into different segments, each of which is mapped to the traffic speed using a linear function.

(4) Predicting traffic speeds. Finally, we combine the K most relevant adjacent roads and their M most relevant traffic attributes to predict the short-term traffic speed of a target road in near future. The parameters used for prediction are obtained through the Particle Swarm Optimization (PSO) algorithm. The traffic speeds are predicted repeatedly once the parameters are determined.

4. Short-term speed forecasting method—NN_Forecast

In this section, we introduce in detail four steps of our proposed short-term speed forecasting approach, i.e., NN_Forecast.

4.1. Finding relevant adjacent roads

This step is employed to find out the K most relevant adjacent roads for each target road with respect to the speed estimation. We use the vehicle trajectory data to measure the degree of correlation between the target road and its surrounding roads, from which the K adjacent roads with largest impact are extracted.

Definition 5 (*Adjacent Road Set and K Most Relevant Roads*)**.** Let $\boldsymbol{nearR}_i = \langle rid_{i,1}, rid_{i,2}, \ldots, rid_{i,T}\rangle$ be a set of adjacent roads of target road r_i in descending order of their relevant weights, in which $rid_{i,j}$ represents the identifier of jth most relevant road to r_i. The corresponding relevant weights of $\boldsymbol{nearR}_i$ is denoted as $\boldsymbol{WR}_i = \langle wr_{i,1}, wr_{i,2}, \ldots, wr_{i,T}\rangle$, in which $wr_{i,j} \geq wr_{i,j+1}$, $1 \leq j \leq T-1$. We define the K most relevant adjacent roads as the roads with top-K corresponding relevant weights in $\boldsymbol{nearR}_i$ which can be denoted as $\boldsymbol{nearR}_i^K = \langle rid_{i,1}, rid_{i,2}, \ldots, rid_{i,K}\rangle$. Correspondingly, the set of relevant weights is denoted as $\boldsymbol{WR}_i^K = \langle wr_{i,1}, wr_{i,2}, \ldots, wr_{i,K}\rangle$. Here, we use the word adjacent to merely mean those surrounding roads whose traffic attributes may influence the future traffic speed of target road r_i. In other words, the road and their adjacent (relevant) roads are not necessarily directly connected geographically.

Definition 6 (*K-Nearest Adjacent Ratio*)**.** Let K-Nearest Adjacent Ratio K-NAR $= \sum_{j=1}^{K} wr_{i,j} / \sum_{j=1}^{T} wr_{i,j}$, which shows the degree of K most relevant adjacent roads on the target road r_i, compared with all T adjacent roads. The greater the ratio, the greater the impact of the K most relevant adjacent roads on the target road r_i.

Suppose that we need to forecast the traffic speeds of each road at time $t+h$ according to the traffic attributes at time t. Given a set of floating vehicle trajectories, ***TP-Set***, for a continuous period of time, Algorithm 1 shows the process of finding the K most relevant adjacent roads. Here, $\boldsymbol{D} = \left[d_{i,j}\right]$ is a $nr \times nr$ matrix, whose element $d_{i,j}$ records the number of *valid* passing vehicles from road r_i to road r_j. Given a predefined time threshold ε_h (e.g., 0.4), if a vehicle passes the road r_i at time t_i, then passes the road r_j at time t_j, and satisfies the formula $\left|\left(t_j - t_i\right)/h - 1\right| < \varepsilon_h$, then we regard the record of the vehicle from road r_i to road r_j as valid, and increase the number of valid passing vehicles from road r_i to road r_j by 1 ($d_{i,j} = d_{i,j} + 1$). The matrix $\boldsymbol{D}$ may also be expressed as a row vector $(\boldsymbol{Y_1}, \boldsymbol{Y_2}, \ldots, \boldsymbol{Y_{nr}})$, in which $\boldsymbol{Y}_j = \left(d_{1,j}, d_{2,j}, \ldots, d_{nr,j}\right)^T$. From matrix $\boldsymbol{D}$, we obtain the adjacent set for any road. For example, to obtain the K adjacent roads of r_j, we select the jth column vector of $\boldsymbol{D}$, i.e., Y_j, and sort the vector elements in descending order. The first K elements of the sorted vector $\left\{d_{p_1,j}, d_{p_2,j}, \ldots, d_{p_K,j}\right\}$ are the K most relevant adjacent roads of r_j.

Algorithm 1 **Finding K most relevant adjacent roads**

Input: Trajectory set ***TP-Set***, road network ***RN***, time span h, time threshold ε_h.
Output: The set of K most relevant adjacent roads of road r_j : $\boldsymbol{nearR}_j^K, j = 1, 2, \ldots, nr$ and the corresponding weight set $\boldsymbol{WR}_j^K,\ j = 1,\ 2, \ldots,\ nr$.

```
D = [d_{i,j}] ← 0 //a nr × nr matrix with each item initialized to zero
for each TP ∈ TP-Set do
  for i ← 1 to TP.length do
    for j ← i + 1 to TP.length do
      if |(TP.P_j.t − TP.P_i.t)/h − 1| < ε_h then
        d_{i,j} = d_{i,j} + 1
      end if
    end for
  end for
end for
for Y_j ∈ D do
  //D = (Y_1, Y_2, ..., Y_nr) and Y_j = (d_{1,j}, d_{2,j}, ..., d_{nr,j})^T
  sort Y_j in descending order, and get the first K elements
    {d_{ρ1,j}, d_{ρ2,j}, ..., d_{ρK,j}} and their corresponding subscripts
    {ρ1, ρ2, ..., ρK}
  nearR_j^K ← {ρ1, ρ2, ..., ρK}
  WR_j^K ← (1 / sqrt(Σ_{i=1}^{K} d²_{ρi,j})) (d_{ρ1,j}, d_{ρ2,j}, ..., d_{ρK,j})  //normalize
    weights
end for
return {nearR_j^K, WR_j^K}_{j=1}^{nr}
```

4.2. Finding relevant traffic attributes

As we have found the K most relevant adjacent roads for each road from the previous step, now we need to find M traffic attributes most relevant to the future traffic speed for each road from its K adjacent roads. We use the gray correlation analysis to measure the degree of correlation between each traffic attribute and future traffic speed, and select the M most relevant traffic attributes. The basic principle of gray correlation analysis is to analyze in a system the degree of similarity between geometric shapes of statistical sequence curves. The more similar the geometric shapes of the sequence curves, the greater the degree of correlation between them. The gray correlation analysis method is applicable to various sample sizes no matter whether the samples are regular or not. In addition, its computations are light-weight with an effective application in the domain of system analysis. Here, we compare the traffic speed–time curves of a road with the traffic attribute-time of its K most relevant adjacent roads. If the shape of a certain sequence curve of one traffic attribute is similar to the sequence curve of traffic speed, the traffic attribute is regarded to be related to the traffic speed.

Definition 7 (*Most Relevant Traffic Attribute Set*)**.** The set of M most relevant traffic attributes for road r_i is denoted as $\boldsymbol{nearS}_i^M = \langle s_{i,1}, s_{i,2}, \ldots, s_{i,M}\rangle$, where $s_{i,j}$ represents the jth traffic attribute most relevant to road r_i. The corresponding weights set of $\boldsymbol{nearS}_i^M$ is denoted as $\boldsymbol{WS}_i^M = \langle ws_{i,1}, ws_{i,2}, \ldots, ws_{i,M}\rangle$, in which $ws_{i,1} \geq ws_{i,2} \geq \cdots \geq ws_{i,M}$.

Based on the idea of gray correlation analysis, we obtain the traffic attributes and their related weights according to the following steps:

(1) **Normalizing traffic speeds and attributes**. The normalized speed of the road r_i at the time point t_z is:

$$nSp_{i,t_z} = \frac{1}{1 + e^{-k_{t_z}\cdot\left(sp_{i,t_z} - \frac{1}{K}\sum_{j=1}^{K} sp_{j,t_z}\right)}} \tag{1}$$

Here, K represents the number of most relevant roads of target road r_i, and sp_{i,t_z} is the speed of the road r_i at the time point t_z. We use the S-curve for data normalization in which k_{t_z} is the parameter that we vary to obtain a better result. In particular, if the variance of traffic attribute values is high, k_{t_z} should be smaller to be more sensitive to these variances. Similarly, if the variance is low, k_{t_z} should be greater. A suggestion to set k_{t_z} is: $k_{t_z} = 10/(q3 - q1)$, where $q1$ and $q3$ are the first quartile and the third quartile of the ordered speed dataset respectively.

The normalized traffic attribute s_m of the road r_i at the time point t_z is:

$$nTs_{i,t_z}(s_m) = \frac{1}{1 + e^{-k_{t_z}(s_m)\cdot\left(ts_{i,t_z}(s_m) - \frac{1}{K}\sum_{j=1}^{K} ts_{j,t_z}(s_m)\right)}} \tag{2}$$

Here, $ts_{i,t_z}(s_m)$ is the traffic attribute s_m of the road r_i at the time point t_z. If one certain traffic attribute is inversely proportional (or negatively correlated) to the traffic speed, we convert its negative correlation into the positive correlation for a fair comparison, i.e., $nTs_{i,t_z}(s_m) = 1 - nTs_{i,t_z}(s_m)$.

(2) **Calculating gray correlation coefficients**. The gray correlation coefficient between each traffic attribute and traffic speed for road r_i is calculated from:

$$r_i(s_m) = \sum_{j=1}^{K}\left(wr_{i,j}\cdot\sum_{z=1}^{p}\left(\Delta_{i,j,t_z}(s_m)\right)^2\right) \tag{3}$$

Here, p is the total number of time slots, and $\Delta_{i,j,t_z}(s_m)$ is the sequence difference value, or $\Delta_{i,j,t_z}(s_m) = 1 - \left|nSp_{i,t_z} - nTs_{j,t_z}(s_m)\right|$.

(3) **Selecting maximum gray correlation degrees**. Select the maximum M gray correlation coefficients for road r_i, $\{r(s'_1), r(s'_2), \ldots, r(s'_M)\}$ and their corresponding traffic attributes $\{s'_1, s'_2, \ldots, s'_M\}$. The gray correlation degree between M traffic attributes and the traffic speed is calculated from:

$$\gamma_i\left(s'_m\right) = \frac{r_i\left(s'_m\right)}{\sum_{j=1}^{M} r_i\left(s'_j\right)} \tag{4}$$

In this way, we obtain the M most relevant traffic attributes of the road r_i: $nearS_i^M = \left(s_{i,1}, s_{i,2}, \ldots, s_{i,M}\right) \Longleftrightarrow (s'_1, s'_2, \ldots, s'_M)$, and the corresponding weights $WS_i^M = \left(ws_{i,1}, ws_{i,2}, \ldots, ws_{i,M}\right) \Longleftrightarrow (\gamma(s'_1), \gamma(s'_2), \ldots, \gamma(s'_M))$.

4.3. Determining interval of traffic attributes

Now, we have found the most relevant traffic attributes of the most relevant adjacent roads for each target road, which can be further converted into the correlation function of traffic speed. However, such correlation is hard to be exactly discovered and also very difficult to use only a single function to reveal the mathematic relationship between traffic attributes and traffic speed, especially when the traffic attribute value is close to saturation. Consequently, we introduce the concept of segmental function. We divide the traffic attribute values into different segments, each of which is mapped to the traffic speed using a linear function. Because for one-dimensional data, Jenks clustering has the advantages of being simple and with higher stability than k-means, we adopt Jenks clustering as the clustering method in this paper.

Jenks clustering groups the data based on their attributes, whose goal is to minimize the difference within the class, and maximize the difference between classes [36]. In a nutshell, the target is to find a clustering that minimizes the sum of variance of each class. The difficulty lies in its poor performance. Therefore, we introduce a dynamic programming approach into Jenks clustering to obtain the optimal solution in less time.

Let $X(s_m) = \left(x_1(s_m), x_2(s_m), \ldots, x_p(s_m)\right)$ be the *Ordered Vector* which contains different values of traffic attribute s_m, collected from all roads at all the time, and $x_j(s_m) < x_{j+1}(s_m)$. Supposing that the attribute value vector $X(s_m)$ is clustered into nc classes $\boldsymbol{C}(s_m) = \{\boldsymbol{C}_1(s_m), \boldsymbol{C}_2(s_m), \ldots, \boldsymbol{C}_{nc}(s_m)\}$, the corresponding splitting points for clustering are then represented by $\boldsymbol{CP}(s_m) = \{cp_1(s_m), cp_2(s_m), \ldots, cp_{nc+1}(s_m)\}$, where $cp_1(s_m) = -\infty, cp_{nc+1}(s_m) = \infty$, and the following formula should be satisfied:

$$\boldsymbol{C}_j(s_m) = \left(x_z(s_m) \,\middle|\, cp_j(s_m) \le x_z(s_m) < cp_{j+1}(s_m),\ \forall z = 1, 2, \ldots, p\right) \tag{5}$$

Because Dynamic Programming (DP) can easily handle nonlinear problems with constraints to obtain a global optimal solution, we solve Jenks clustering through DP as follows.

(1) **Analyzing the structure of the optimal solution**. The Jenks clustering problem of an ordered attribute value vector with nx values clustered into nc classes can be abbreviated as $\mathbf{X}[1: nx;\ nc]$. We need to find out the optimal intervals of clustering. Suppose two clusters are disconnected between attribute values x_v and x_{v+1} ($v = 1, 2, \ldots, nx-1$), i.e., x_v and x_{v+1} belong to different classes. The total variance of $\mathbf{X}[1: nx;\ nc]$ is the sum of the variance of $\mathbf{X}[1: v;\ nv]$ and the variance of $\mathbf{X}[v+1: nx;\ nc-nv]$.

Theorem 1. *The sub-clusters* $\mathbf{X}[1: v;\ nv]$ *and* $\mathbf{X}[v+1: nx;\ nc-nv]$ *included in the optimal clustering of* $\mathbf{X}[1: nx;\ nc]$ *are also optimal clustering.*

Proof of Theorem 1. Suppose $\mathbf{X}[1: nx;\ nc]$ is the optimal clustering, $\mathbf{X}[1: v;\ nv]$ is a sub-cluster of $\mathbf{X}[1: nx;\ nc]$. If there exists another partition $\mathbf{X}'[1: v;\ nv]$ whose variance is less than $\mathbf{X}[1: v;\ nv]$, we can substitute this cluster for the original cluster, to make the sum of the variance of the clustering $\mathbf{X}[1: nx;\ nc]$ less, which leads to a contradiction with the assumption that $\mathbf{X}[1: nx;\ nc]$ is an optimal clustering.

According to Theorem 1, the optimal solution of Jenks clustering problem contains the optimal solution of its sub-problems.

(2) **Establishing recursive relations**. For the Jenks clustering problem, suppose the minimum variance of $\mathbf{X}[1: i;\ j]$ is $\boldsymbol{SV}[i][j]$, and the last splitting point is $\boldsymbol{BP}[i][j]$ ($\boldsymbol{BP}[i][j]-1$ and $\boldsymbol{BP}[i][j]$ belong to different classes), thus the minimum variance of the original problem is $\boldsymbol{SV}[nx][nc]$, and the last splitting point is $\boldsymbol{BP}[nx][nc]$.

When $i = 1$, regardless of the value of j, $\boldsymbol{SV}[i][j] = 0$, $\boldsymbol{BP}[i][j] = 1$.

When $i \ge j$, then $\boldsymbol{SV}[i][j] = \boldsymbol{SV}[\boldsymbol{BP}[i][j]][j-1] + \sigma^2(\boldsymbol{BP}[i][j]: i)$. Here, $\sigma^2(\boldsymbol{BP}[i][j]: i)$ is the variance of sequence $x_{\boldsymbol{BP}[i][j]}, x_{\boldsymbol{BP}[i][j]+1}, \ldots, x_i$. When the value $\boldsymbol{BP}[i][j]$ is undefined, $\boldsymbol{SV}[i][j]$ is calculated recursively as the following:

$$\boldsymbol{SV}[i][j] = \left\{\boldsymbol{SV}[k-1][j-1] + \sigma^2(k: i)\right\} \tag{6}$$

(3) **Constructing the optimal clustering**. For Jenks clustering problem of $\mathbf{X}[1: nx;\ nc]$, firstly we obtain the last class $\boldsymbol{C}_{nc} = \left\{x_{p_{nc}}, x_{p_{nc}+1}, \ldots, x_{nx}\right\}$ according to the splitting point $\boldsymbol{BP}[nx][nc]$, also recorded as p_{nc}. For sub-clustering problem of $\mathbf{X}[1: p_{nc}-1;\ nc-1]$, it is also an optimal clustering according to Theorem 1. Afterwards, we obtain the penultimate class $\boldsymbol{C}_{nc-1} = \left\{x_{p_{nc-1}}, x_{p_{nc-1}+1}, \ldots, x_{p_{nc}-1}\right\}$ according to the splitting point $\boldsymbol{BP}[p_{nc}-1][nc-1]$, also recorded as p_{nc-1}. Through the iterations, we obtain the optimal clustering $\boldsymbol{C} = \{\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_{nc}}\}$, and its corresponding splitting points for clustering $\boldsymbol{CP} = \left\{-\infty, x_{p_2}, \ldots, x_{p_{nc}}, \infty\right\} \Longleftrightarrow \{cp_1, cp_2, \ldots, cp_{nc+1}\}$.

Algorithm 2 Jenks Clustering using dynamic programming

Input: class number nc, Ordered Vector $\boldsymbol{X} = (x_1, x_2, \ldots, x_{nx})$.
Output: the optimal clustering $\boldsymbol{C}$, and its corresponding splitting points for clustering $\boldsymbol{CP}$.

1: $\boldsymbol{BP}, \boldsymbol{SV} \leftarrow$ a $nx \times nc$ matrix with each item initialized to zero
2: // initialize $\boldsymbol{BP}$ and $\boldsymbol{SV}$
3: **for** $i \leftarrow 1$ ***to*** nc **do**
4: $\quad\boldsymbol{BP}[1][i] = 1, \boldsymbol{SV}[1][i] = 0$
5: $\quad$**for** $j \leftarrow 2$ ***to*** nx **do**
6: $\quad\quad\boldsymbol{BP}[j][i] = \infty$
7: $\quad$**end for**
8: **end for**
9: // obtain $\boldsymbol{BP}$ and $\boldsymbol{SV}$
10: **for** $xNum \leftarrow 2$ ***to*** nx **do**
11: $\quad$**for** $lastN \leftarrow 1$ ***to*** $xNum$ **do**
12: $\quad\quad va \leftarrow$ the variance of $\boldsymbol{X}$ from $lastN$ to $xNum$, taking the number for each value into account
13: $\quad\quad$**if** $lastN\ != xNum$ **then**
14: $\quad\quad\quad$**for** $cNum \leftarrow 2$ ***to*** nc **do**
15: $\quad\quad\quad\quad$**if** $\boldsymbol{SV}[xNum][cNum] >= va + \boldsymbol{SV}[xNum - lastN][cNum-1]$ **then**
16: $\quad\quad\quad\quad\quad\boldsymbol{BP}[xNum][cNum] = xNum - lastN + 1$
17: $\quad\quad\quad\quad\quad\boldsymbol{SV}[xNum][cNum] = va + \boldsymbol{SV}[xNum - lastN][cNum-1]$
18: $\quad\quad\quad\quad$**end if**
19: $\quad\quad\quad$**end for**
20: $\quad\quad$**end if**
21: $\quad$**end for**
22: $\quad\boldsymbol{BP}[xNum][1] = 1,\ \boldsymbol{SV}[xNum][1] = va$
23: **end for**
24: // Obtain the optimal solution by $\boldsymbol{BP}$
25: $\boldsymbol{cBP} \leftarrow$ a $nc+1$ vector, $\boldsymbol{CP} \leftarrow$ a $nc+1$ vector, $\boldsymbol{C} \leftarrow$a nc vector
26: $count = nc+1,\ \boldsymbol{cBP}[1] = 1,\ \boldsymbol{cBP}[nc+1] = nx+1,\ \boldsymbol{CP}[1] = -\infty,\ \boldsymbol{CP}[nc+1] = \infty$
27: **while** $--count >= 2$ **do**
28: $\quad idx = \boldsymbol{cBP}[count+1] - 1$
29: $\quad\boldsymbol{CP}[count] = x_{idx}$
30: $\quad\boldsymbol{cBP}[count] = \boldsymbol{BP}[idx][count] - 1$
31: **end while**
32: **for** $i \leftarrow 1$ ***to*** nc **do**
33: $\quad$**for** $j \leftarrow \boldsymbol{cBP}[i]$ ***to*** $\boldsymbol{cBP}[i+1]-1$ **do**
34: $\quad\quad\boldsymbol{C}[i] = \boldsymbol{C}[i] \cup x_j$
35: $\quad$**end for**
36: **end for**
37: **return** $\boldsymbol{C}$, $\boldsymbol{CP}$

Algorithm 2 gives the complete process of solving the optimization problem of Jenks clustering using DP. Lines 3 to 8 show the initialization of $\boldsymbol{BP}$ and $\boldsymbol{SV}$, whereas Lines 10 to 23 show the process of solving $\boldsymbol{BP}$ and $\boldsymbol{SV}$ in a bottom-up manner. The process of constructing optimal clustering by $\boldsymbol{BP}$ is given from Lines 25 to 36.

The solving process of Jenks clustering problem using the exhaustive method is equivalent to finding $nx-1$ values from $nc-2$ values as clustering points, leading to $\binom{nc-2}{nx-1}$ cases. Thus, the time complexity of exhaustive method is $O(nc^{nx})$. In contrast, the main steps involved in DP is the calculation of $\boldsymbol{BP}$ and $\boldsymbol{SV}$, whose time complexity is $O\left(nx \cdot nc^2\right)$. To Solve Jenks clustering problem by using Dynamic Programming (DP) can not only spend less running time, but also obtain a good clustering effect when a large number of data are divided into only a few classes.

Besides, the ***SV*** calculated by DP contains the minimum variance of the first i items clustered from any j classes of the attribute vector $\mathbf{X}$, whereas ***BP*** holds the clustering points with minimum variance. When we are not sure how many classes to cluster in, we can set the maximum number of classes, calculate the relationship between the number of classes and the minimum variance, to determine the optimal number of classes and construct the optimal clustering according to***BP***.

4.4. Predicting short-term traffic speeds

In step 3, we have used Jenks clustering to determine the segment intervals of the linear piecewise function of traffic attributes of the adjacent roads and the future traffic speed. In this step, we combine the adjacent roads and relevant traffic attributes to predict the traffic speed, and determine the parameters of piecewise function.

The correlation function of the traffic attribute in each segment interval for the road r_i satisfies the linear relation. We use $f(x; a, b) = ax + b$ to represent a segment of the segment function. Therefore, the mth relevant traffic attribute of road r_i in the kth interval corresponds to the linear function $f\left(x; a_{i,K}\left(s_{i,m}\right), b_{i,K}\left(s_{i,m}\right)\right)$, abbreviated as $f_{i,k,m}(x)$. If the value of an associated traffic attribute v lies in the range of $\left[cp_k\left(s_{i,m}\right), cp_{k+1}\left(s_{i,m}\right)\right]$, and its corresponding value is $f_{i,k,m}(v)$, then the corresponding function is defined as:

$$g_{i,m}(v) = \sum_{k=1}^{nc}\left(l\left(cp_k\left(s_{i,m}\right) \le v < cp_{k+1}\left(s_{i,m}\right)\right) \cdot f_{i,k,m}(v)\right) = \sum_{k=1}^{nc}\left(l\left(v \in \boldsymbol{C}_k\left(s_{i,m}\right)\right) \cdot f_{i,k,m}(v)\right) \tag{7}$$

Here, $l(\cdot)$ is an indication function which gives the value of 1 when the parameter is true, otherwise 0, i.e., $l(True) = 1,\ l(False) = 0$.

Then, we obtain the contribution of the traffic attribute $s_{i,m}$ to the future short-term traffic speed for road r_i:

$$\overline{\overline{sp_{i,t_z+h(s_{i,m})}}} = \sum_{j=1}^{K} wr_{i,j} \cdot g_{i,m}\left(ts_{j,t_z}\left(s_{i,m}\right)\right) \tag{8}$$

Here, $wr_{i,j}$ is the influence weight of adjacent road r_j for the target road r_i, and $ts_{j,t_z}\left(s_{i,m}\right)$ is the value of traffic attribute $s_{i,m}$ of the jth adjacent road on target road r_i at time point t_z.

Thus, we predict the traffic speed of the road r_i at the time point $t_z + h$ by the relevant traffic attributes of the adjacent roads:

$$\overline{\overline{sp_{i,t_z+h}}} = \sum_{m=1}^{M} ws_{i,m} \cdot \overline{\overline{sp_{i,t_z+h(s_{i,m})}}} \tag{9}$$

For target road r_i, the objective function is defined as:

$$R_i = \frac{\lambda}{2} \cdot \sum_{m=1}^{M}\left(a_{i,1}^2\left(s_{i,m}\right) + \cdots + a_{i,nc}^2\left(s_{i,m}\right) + b_{i,1}^2\left(s_{i,m}\right) + \cdots + b_{i,nc}^2\left(s_{i,m}\right)\right) + \frac{1}{2p} \cdot \sum_{z=1}^{p}\left(sp_{i,t_z+h} - \overline{\overline{sp_{i,t_z+h}}}\right) \tag{10}$$

Here, p is the total number of time slots, and λ is the regularization parameter used to avoid overfitting. We need to find the model parameter vector $\theta_i = \left(a_{i,1}\left(s_{i,1}\right),\ a_{i,2}\left(s_{i,1}\right), \ldots,\ a_{i,nc}\left(s_{i,M}\right),\ b_{i,1}\left(s_{i,1}\right),\ b_{i,2}\left(s_{i,1}\right), \ldots, b_{i,nc}\left(s_{i,M}\right)\right)^T$ which makes the objective value of function (10) minimum.

We employ the Particle Swarm Optimization (PSO) algorithm to solve the parameter values, which first initializes a population of random particles and then iteratively finds the optimal solution.

Table 1
Parameters of PSO.

Swarm size	50
Max iterations (max_iter)	200
Inertia weight of the ith iteration ($\omega(i)$)	$\omega(i) = 0.9 - 0.7i/max\_iter$
Acceleration coefficient (c_1)	$c_1 = 2.5 - 2i/max\_iter$
Acceleration coefficient (c_2)	$c_2 = 0.5 + 2i/max\_iter$

Table 2
Examples of GPS sampling records.

Plate number	Latitude	Longitude	Speed (km/h)	Direction	Sampling time
00001	120.62625	28.00968	39	270	2015/2/1 00:03:59
00002	120.78025	27.94367	76	45	2015/2/1 00:03:59
00003	120.65142	27.99157	24	225	2015/2/1 00:04:10
...	...	...	...	...	...

By tracking both the optimal location p_{best} of each particle found by the current iteration and the global optimum location g_{best} of all particles, we update their speed and location according to:

$$V_k = \omega \cdot V_{k-1} + c_1 \cdot r(\cdot) \cdot \left(p_{best} - p_{present}\right) + c_2 \cdot r(\cdot) \cdot \left(g_{best} - p_{present}\right) \tag{11}$$

$$p_{present_k} = p_{present_{k-1}} + V_k \tag{12}$$

where V represents the speed of the particle, $p_{present}$ the current location of the particle, $r(\cdot)$ a random number between (0,1), c_1 and c_2 the learning factors and ω the inertia factor with a value in the range of 0.1 to 0.9. Table 1 presents the key parameters of PSO according to the work given by Zhang et al. [37].

5. Experiment

5.1. Preparation

In order to evaluate the effectiveness of our approach, we conducted extensive experiments in Wenzhou, a major city in eastern China with a population of more than nine million. The road network ranges in Wenzhou between 120.5–120.9E and 27.8–28.15N which covers 43,038 roads.

We investigated the real 37,884,970 GPS records of 3743 moving vehicles, collected from February 1, 2015 to February 3, 2015, with 12,727,709 GPS records on first day, 12,572,842 GPS records on the second day, and 12,584,419 GPS records on the third day. As shown in Table 2, each GPS record includes the plate number, the position (latitude and longitude), the moving speed, the moving direction, and the time of GPS record sampled. The interval of two consecutive GPS sampling time varies from 1 to 30 s. However, the sampling interval remains the same for the same vehicle. We divide the whole day equally into 288 (5-min each), 144 (10-min each) or 96 (15-min each) time slots respectively. Therefore, each GPS record can be allocated to a fixed time slot. For a certain time slot and a certain road, we use the average speed calculated from the past GPS records sampled on this road as the moving speed of this road over this time slot. Similarly, we obtain the traffic flow for each road and each time slot. Furthermore, by combining the road capacity and road length, we also obtain the road occupancy and traffic density for each road and each time slot.

During our experiment, we select the data of first two days as the model-training data, and the data of the third day as the testing data to evaluate our approach. When we predict the traffic speed $sp_{i,t}$ of road r_i in time slot t on the third day, we first find its adjacent roads set $\boldsymbol{nearR}_i^K = \left(rid_{i,1}, rid_{i,2}, \ldots, rid_{i,K}\right)$ of road r_i. Based on the traffic attribute values of these adjacent roads in the previous time slot, i.e., the time slot $t - h$, we then calculate the predicted traffic speed $\overline{\overline{sp_{i,t}}}$ by our proposed forecasting approach.

5.1.1. Evaluation methods

To show the advantages of our NN_Forecast algorithm over the traditional short-term traffic forecasting algorithms, we compare our results with that of three state-of-the-art algorithms, i.e., improved KNN, C-WSVM and ARIMA. We select these three algorithms for comparison not only because they also focus on short-term traffic speed forecasting, but also because their implementation can be fulfilled according to the references.

(1) **Improved KNN**. The K-Nearest-Neighbors algorithm (or KNN for short) is a non-parametric method used for classification and regression. In KNN classification, an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. KNN algorithm is one of the oldest and simplest learning algorithm, and its effectiveness has been proven in many papers. In this paper, an improved method of KNN [23] is compared, which considers the influence of the adjacent roads. This improved KNN has three major parameters: the sequence length, the number of neighbors k, and the number of adjacent roads. As the experiment in [23] suggests, we set the sequence length to 6, the number of neighbors k to 1, and the number of adjacent roads to the same as those used in our experiment.

(2) **C-WSVM**. C-WSVM is a short-term traffic forecasting hybrid model proposed by Wang et al. [28]. On the basis of the original SVM model, it employs the wavelet function as the kernel of the model, and determines the dimension of the input space by the phase space reconstruction theory. In our experiment, we set the parameter of delay τ equal to the interval of time slots in our approach, the sequence length m to 6, tolerable deviation ε to 0.001, the penalty coefficient C (classification error) to 1000, and the kernel function parameter a to 6.

(3) **ARIMA**. As a well-known time series prediction method, the effectiveness of Auto-Regressive Integrated Moving Average (ARIMA) has been verified in many scenarios. For the order (number of time lags) of the autoregressive model p and the order of the moving-average q, we first set a series of parameters in advance, such as $(p, q) = (1, 2)$ and $(p, q) = (2, 1)$, then calculated the Akaike Information Criterion (AIC) of the model using these different parameters with the real data, and finally chose the pair of parameters which had the minimum AIC. For the degree of differencing d, we tested its effectiveness when d was set to 0, 1, 2 or 3, based on the real data, and found the best one when $d = 1$. In our experiment, we set the parameter of the training set length T to 15, the parameter p to 0, the degree of differencing d to 1, and the parameter q to 1.

5.1.2. Evaluation metrics

We adopt two popular metrics to evaluate the performance of NN_Forecast, namely, Mean-Absolute-Percentage Error (MAPE) and Root-Mean-Square Error (RMSE). The smaller MAPE or RMSE value means better prediction accuracy. In the following equations, $\overline{\overline{sp_i}}$ is the predicted traffic speed and sp_i is the actual traffic speed.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|sp_i - \overline{\overline{sp_i}}\right|}{sp_i} \times 100\% \tag{13}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(sp_i - \overline{\overline{sp_i}}\right)^2}{N}} \tag{14}$$

MAPE reveals the degree of deviation between the actual and predicted values, while RMSE is sensitive to extremely large or small errors in a set of measurements and is good at measuring precision.

Meanwhile, in order to analyze the difference between the predicted value and the actual value in detail, we also consider the proportion of difference between the predicted value and the actual value, such as that of 2 km/h, 3 km/h, 4 km/h, 5 km/h and 10 km/h.

Table 3
The influence of the number of the nearest adjacent roads on the nearest adjacent ratio for different time intervals.

Time interval (min)	K-Nearest adjacent ratio	K = 1	K = 2	K = 3	K = 4	K = 5	K = 6
5	[0, 0.5)	25.1%	10.6%	5.1%	2.5%	1.4%	0.8%
	[0.5, 0.6)	16.0%	6.2%	3.9%	2.9%	1.4%	0.9%
	[0.6, 0.7)	4.2%	8.6%	4.0%	3.0%	2.4%	1.7%
	[0.7, 0.8)	2.0%	2.9%	5.6%	2.8%	2.5%	2.5%
	[0.8, 1.0]	52.7%	71.7%	**81.4**%	88.7%	92.2%	94.0%
10	[0, 0.5)	34.5%	19.0%	11.5%	7.5%	5.2%	3.8%
	[0.5, 0.6)	14.8%	6.8%	5.1%	4.4%	3.1%	2.1%
	[0.6, 0.7)	4.0%	9.0%	5.5%	4.5%	3.3%	3.2%
	[0.7, 0.8)	2.3%	3.0%	5.5%	2.8%	3.8%	3.4%
	[0.8, 1.0]	44.5%	62.2%	72.5%	**80.9**%	84.6%	87.6%
15	[0, 0.5)	39.1%	23.6%	15.8%	11.5%	8.5%	6.6%
	[0.5, 0.6)	14.5%	6.8%	5.7%	5.0%	3.8%	2.7%
	[0.6, 0.7)	4.3%	9.1%	5.1%	4.9%	3.7%	3.9%
	[0.7, 0.8)	2.3%	3.6%	6.0%	2.9%	4.3%	3.9%
	[0.8, 1.0]	39.8%	57.0%	67.5%	75.7%	79.7%	**82.8**%

5.2. Data profile and parameter setting

5.2.1. Data profile

Fig. 2 shows the average traffic speed over time from February 1 to February 3, 2015 (from Sunday to Tuesday) in Wenzhou. It can be seen that the rush hours occur approximately from 05:00 to 10:00 in the morning and from 19:00 to 21:30 in the evening for all three days. Although the change of traffic speed during three days shares similar characteristics roughly, the rush hours in Sunday evening are less obvious, compared with those in evenings of the other two days. Furthermore, the average speed on Sunday achieves 28.4 km/h, whereas the average speeds of two working days achieve 27.3 km/h and 27.1 km/h respectively. Hence, we can draw the conclusion that vehicles are driven faster on days off than on working days, which is consistent with the real case.

5.2.2. Varying the number of nearest adjacent roads (k)

According to Algorithm 1 presented in Section 4.1, we obtain the distribution of *K-Nearest Adjacent Ratio* with respect to the number of adjacent roads at different ratio intervals and time intervals, as shown in Table 3. The value in the table indicates the proportion of the roads whose *K-Nearest Adjacent Ratios* fall in the corresponding ratio interval (indicated by the second left column) with the certain number of the nearest adjacent roads (indicated by the first row) and the certain time interval (indicated by the first left column). For example, with only one nearest adjacent road ($K = 1$), the *K-Nearest Adjacent Ratios* of 25.1% roads fall in [0, 0.5].

From Table 3 we conclude that the greater the time interval, the more number of nearest adjacent roads which would impact on the traffic speed, leading to more difficulties if using only a small number of adjacent roads to predict future traffic speed. When the number of nearest adjacent road is set to 2, i.e., $K = 2$, the *K-Nearest Adjacent Ratios* of more than half roads fall in [0.8, 1.0], which indicates that most of the vehicles on the road are driven from a small number of adjacent roads. If the proportion of *K-Nearest Adjacent Ratio* falling in [0.8, 1.0] should be greater than 80%, then the smallest number of nearest adjacent roads should be set to 3 for time interval of 5 min, 4 for time interval of 10 min, and 6 for time interval of 15 min, as shown in the bold figures in Table 3.

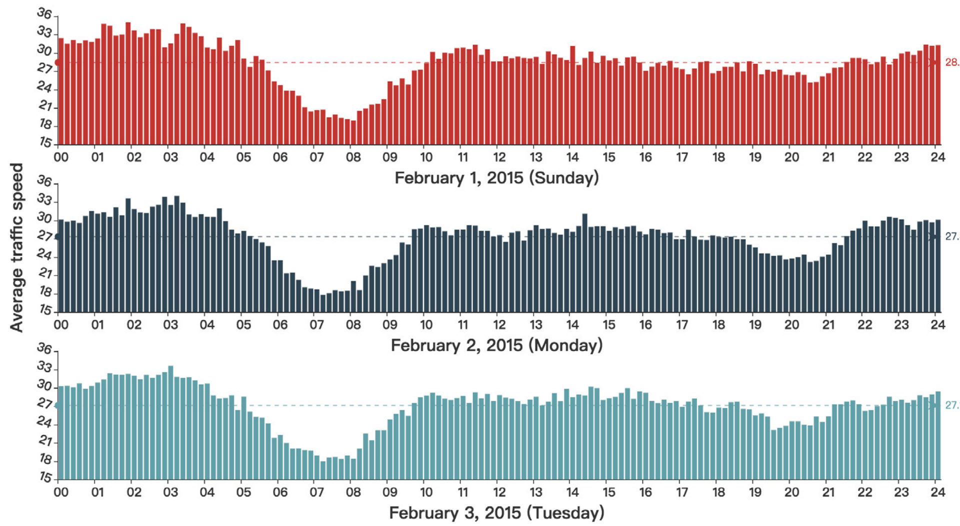


Fig. 2. The change of average traffic speed with time.

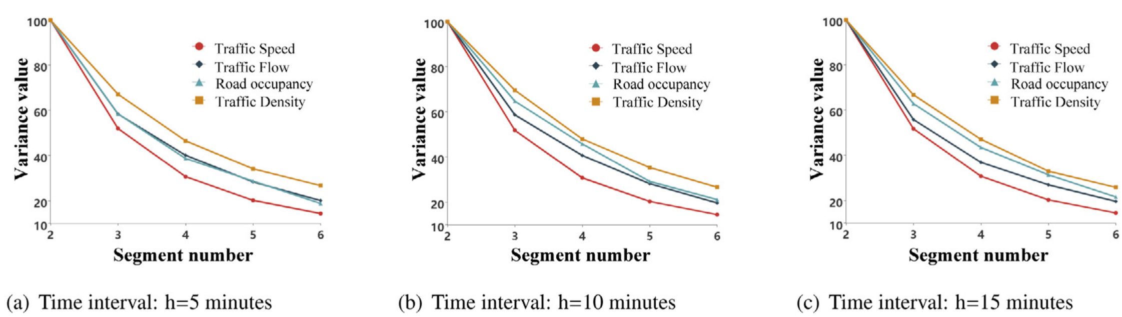


Fig. 3. The influence of different segment numbers on cluster variance.

5.2.3. Varying the number of segments (nc)

The values of four traffic attributes are clustered according to the Jenks clustering method with dynamic programming given in Section 4.3. Fig. 3 shows the variances of four traffic attributes with different numbers of segment and time intervals. In order to facilitate the comparison of variances of different traffic attributes, we normalize the variances as following:

$$\mathrm{VA}'_j(s_m) = 100 \cdot \frac{\mathrm{VA}_j(s_m)}{\mathrm{VA}_2(s_m)} \tag{15}$$

in which $\mathrm{VA}_j(s_m)$ represents the variance of the traffic attribute s_m with j segments, and $\mathrm{VA}'_j(s_m)$ is the normalized variance of $\mathrm{VA}_j(s_m)$.

It can be seen from Fig. 3 that the variance curves of different traffic attributes are close to each other, always with the largest variance of traffic speed, followed by traffic flow, road occupancy, and traffic density, regardless of time intervals. In addition, only the variance of road occupancy is significantly different at different time intervals. When the segment number nc increases from 2 to 5, the variances gradually decrease no matter what the time interval is, which means the segmentation effect is improved gradually. However, when nc continues to increase from 5, the variances turn out to be more stabilized. Furthermore, the more number of segments requires the more model parameters, leading to more time to build the model. Therefore, we choose $nc = 5$ at the intervals of 5-min, 10 min and 15 min.

Based on the above analysis, we obtain the following parameters: $K = 3$ (number of adjacent roads), $nc = 5$ (number of segments) at 5-min interval; $K = 4$, $nc = 5$ at 10-min interval; and $K = 6$, $nc = 5$ at 15-min interval.

5.3. Experimental results

5.3.1. Comparative results

In order to compare the prediction results of three approaches at different time points, we divide the whole day of February 2 into 12 slots, with each 2 h. Fig. 4 shows the box plot of the differences between the actual speed and predicted speed over 5 time slots. As can be seen from the figure, although the differences of all four approaches almost fall in the range of 5–10 km/h, NN_Forecast has the highest stability, followed by ARIMA. In addition, NN_Forecast has the lowest mean difference than the other three approaches for all time slots.

Table 4 compares the prediction results of the approaches at different time intervals, in which the best ones are presented in bold. It can be easily found that with the increase of time interval, MAPE and RMSE of NN_Forecast, Improved KNN and C_WSVM all increase, with those of NN_Forecast increasing most obviously. To predict the traffic speed in 5-min and 10-min, NN_Forecast outperforms the other three approaches. As for the 15-min prediction, however, NN_Forecast is significantly less predictive than ARIMA. It shows that NN_Forecast has the high accuracy when predicting the short-term speed, whereas ARIMA is least affected by the time interval.

To further investigate the effectiveness of NN_Forecast, we divided the period from 7:00 to 22:00 on February 2 2015 into 180 time slots, with each having 5 min, and performed the significance tests on MAPEs and RMSEs of NN_Forecast with those of improved KNN, C-WSVM and ARIMA for 5-min prediction with all time slots. As Table 5 indicates, in the case of the significance level being set to 0.1, i.e., $\alpha = 0.1$, we can statistically accept that NN_Forecast

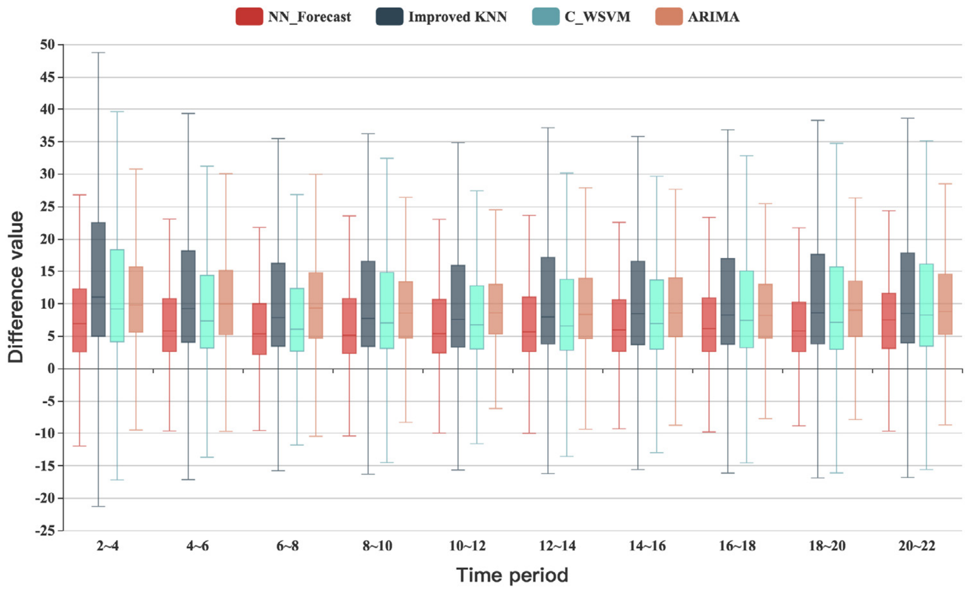


Fig. 4. Box plot for the difference between predicted and actual speed for different time points (10-min interval).

Table 4
Comparison of speed prediction accuracy.

Time interval (min)	Methods	≤2 km/h	≤3 km/h	≤4 km/h	≤5 km/h	≤10 km/h	MAPE	RMSE
5	NN_Forecast	**21.32%**	**30.58%**	**39.31%**	**47.56%**	**74.65%**	**18%**	**10.55**
	Improved KNN	15.75%	23.22%	30.36%	36.93%	61.57%	32%	16.05
	C-WSVM	18.79%	27.55%	35.56%	43.02%	69.00%	27%	13.25
	ARIMA	13.37%	21.87%	29.28%	36.46%	63.59%	34%	16.83
10	NN_Forecast	**17.34%**	**25.32%**	**33.20%**	**40.45%**	**65.76%**	**26%**	**12.13**
	Improved KNN	14.45%	21.40%	28.04%	34.30%	58.73%	36%	17.43
	C-WSVM	14.76%	21.93%	28.51%	34.43%	59.54%	28%	15.24
	ARIMA	14.81%	22.02%	28.88%	36.41%	62.93%	27%	14.86
15	NN_Forecast	13.74%	20.61%	25.79%	31.33%	53.24%	39%	17.71
	Improved KNN	12.25%	18.19%	23.99%	29.43%	52.09%	44%	20.38
	C-WSVM	13.50%	20.13%	26.45%	32.45%	56.69%	35%	17.38
	ARIMA	**15.53%**	**22.98%**	**30.22%**	**37.00%**	**64.16%**	**26%**	**12.11**

Table 5
Results of significance test.

	MAPE	RMSE
p-value: NN_Forecast vs. Improved KNN	0.0185	0.0370
p-value: NN_Forecast vs. C-WSVM	0.0787	0.0963
p-value: NN_Forecast vs. ARIMA	0.0324	0.0688

outperforms improved KNN, C-WSVM and ARIMA with regard to MAPE and RMSE for 5-min prediction.

5.3.2. Impact of different traffic attributes

For each time interval, according to the steps to find the relevant traffic attributes in Section 4.2, we obtain various indicator values of traffic attributes. We choose four traffic attributes ($M = 4$): traffic speed, traffic flow, road occupancy and traffic density. For each target road r_i, we obtain its most relevant traffic attributes set $\boldsymbol{nearS}_i^M = (s_{i,1}, s_{i,2}, \dots, s_{i,M})$, $s_{i,j} \in \{trafficspeed, trafficflow, roadoccupancy, trafficdensity\}$, $1 \le j \le M$, and correspondingly $\boldsymbol{WS}_i^M = (ws_{i,1}, ws_{i,2}, \dots, ws_{i,M})$, $\sum_{j=1}^{M} ws_{i,j} = 1$. Then we obtain the influence degree corresponding to traffic flow:

$$InfluenceDegree_{traffic\_flow} = \frac{1}{nr}\sum_{i=1}^{nr}\sum_{j=1}^{M} l\left(s_{i,j} = traffic\_flow\right) \cdot ws_{i,j} \tag{16}$$

Here, *InfluenceDegree* indicates the extent that how much the traffic attribute would influence the future traffic speed.

For each road r_i, $s_{i,1}$ holds the most relevant traffic attribute with the future traffic attribute. Then we obtain the most relevant attribute ratio corresponding to traffic flow:

$$MostRelevant_{traffic\_flow} = \frac{1}{nr}\sum_{i=1}^{nr} l\left(s_{i,1} = traffic\_flow\right) \tag{17}$$

Here, *MostRelevant* indicates the percentage of roads whose most relevant traffic attribute to future traffic speed is the corresponding traffic attribute.

For a certain road, we do not know whether each traffic attribute of the adjacent roads is positively or negatively correlated with the future traffic speed of the road beforehand. Hence, we calculate both the positive correlation weight and the negative correlation weight respectively. If the positive correlation weight value is larger than the negative correlation weight value, we consider that the traffic attribute is positively related to the traffic speed. Otherwise, we consider a negative correlation. Here, we set $ws_{i,j} = \{ws_{i,j}^+, ws_{i,j}^-\}$, in which $ws_{i,j}^+$ represents positive correlation coefficient, and $ws_{i,j}^-$ represents negative correlation coefficient. For example, given the future traffic speed of 0.8 (after the normalization) and traffic flow of 0.2, we have $ws^+ = 1 - |0.8 - 0.2| = 0.4$. When the negative correlation value is calculated, the traffic flow rate is actually changed to $1 - 0.2 = 0.8$, and we have $ws^- = 1 - |0.8 - 0.8| = 1$. At this time $ws^- > ws^+$ (1 >

Table 6
The influence of relevant traffic attribute.

Time interval (min)	Traffic attribute	Influence degree	Most relevant attribute ratio	Positive correlation ratio	Negative correlation ratio
5	Traffic speed	26.5%	54.0%	77.6%	22.4%
	Traffic flow	24.7%	17.7%	29.7%	70.3%
	Road occupancy	24.4%	15.1%	29.7%	70.3%
	Traffic density	24.4%	13.2%	29.7%	70.3%
10	Traffic speed	26.1%	52.2%	73.9%	26.1%
	Traffic flow	24.6%	17.6%	27.0%	73.0%
	Road occupancy	24.6%	15.0%	26.8%	73.2%
	Traffic density	24.6%	15.2%	26.6%	73.4%
15	Traffic speed	26.3%	53.4%	77.2%	22.8%
	Traffic flow	24.6%	17.1%	30.9%	69.1%
	Road occupancy	24.5%	14.9%	30.9%	69.1%
	Traffic density	24.6%	14.6%	31.1%	68.9%

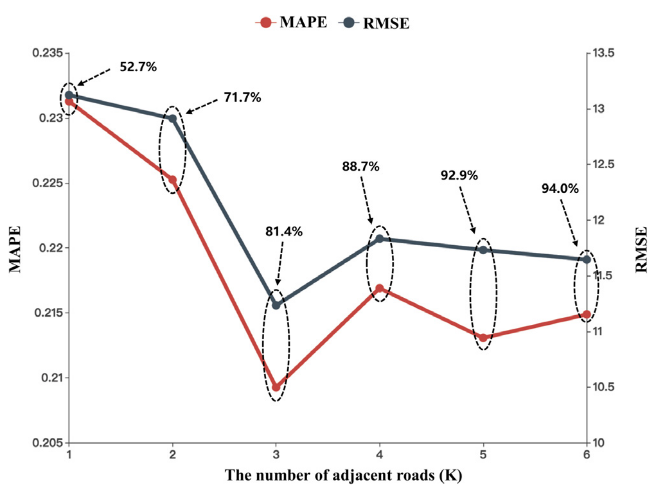


Fig. 5. Impact of parameter K on prediction accuracy (5-min interval).

0.4). Consequently, for the road r_i, the traffic flow is negatively correlated.

As Table 6 indicates, the numerical values are very close no matter what the time intervals are set to. The influence degrees of four traffic attributes on future traffic speed are almost equal, although that of traffic speed is slightly higher. The traffic speed of nearly half of the roads are most closely related to the future traffic speeds, with the traffic flow following. For the majority of roads, their traffic speeds are positively related with their future traffic speeds ($ws_{i,j}^{+} > ws_{i,j}^{-}$). However, for more roads, their attributes of the traffic flow, road occupancy and traffic density have the negative correlations with their future traffic speed ($ws_{i,j}^{-} > ws_{i,j}^{+}$).

5.3.3. Impact of parameter K on prediction accuracy

Fig. 5 illustrates the change of MAPE and RMSE with K, i.e., the number of adjacent roads, at the time interval of 5 min for NN_Forecast. As it shows, the prediction accuracy achieves the best when $K = 3$. In addition, when $K \geq 4$, the prediction accuracy keeps steady and changes little. In other words, selecting too few adjacent roads certainly deteriorates its performance as we expect. However, selecting too many adjacent roads does not promote its performance too much. The tag labeled on the lines represents the proportion of *K-Nearest Adjacent Ratio* falling in [0.8, 1.0] under the corresponding number of adjacent roads.

5.3.4. Impact of parameter nc on prediction accuracy

Fig. 6 presents the performance of NN_Forecast by evaluating the MAPE and RMSE with different *nc*, i.e., the number of clusters

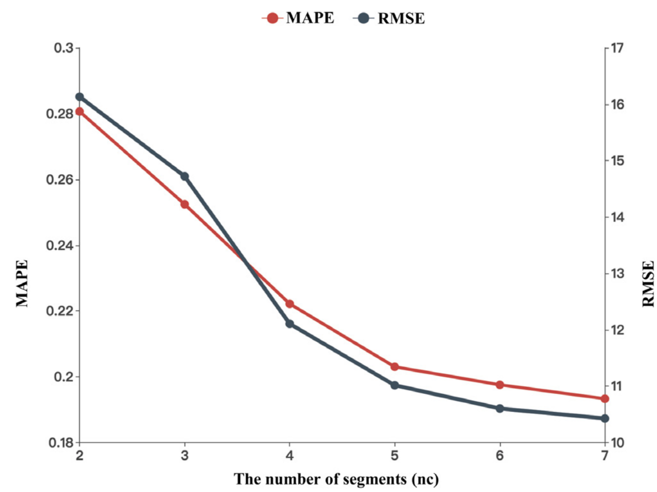


Fig. 6. Impact of parameter nc on prediction accuracy (5-min interval).

(or segments), at 5-min interval. As shown in Fig. 6, both the MAPE and RMSE values decrease gradually with more number of segments. However, when *nc* continues to increase, the prediction accuracy tends to be stable.

5.3.5. Some insights from real data

In order to more intuitively view the process and effect of forecasting speed, we select part of the roads in Wenzhou City on 7:05 02-02-2015 (morning peak period) for speed prediction. Fig. 7(a) and (b) demonstrate the real speeds of these roads for the time slots of 7:00 and 7: 05 respectively, whereas Fig. 7(c) shows the prediction speed for the time slot of 7: 05. Each road is labeled as the format of *road-identifier (speed)*. By comparing Fig. 7(b) and (c), we conclude that the results of the predicted speed and the real speed are basically the same (the error almost fall in the range of 5–10 km/h). Besides, Fig. 7 also reveals the dynamics of traffic. The vehicles traveling on *Road 21312* are mainly from *Road 21310* and *21311*, with their *K-Nearest Adjacent Ratios* of 0.47 and 0.25 respectively (including the periods of whole day), which reveals a dominant driving route of the vehicles, i.e., 21 310 → 21 311 → 21 312. Meanwhile, the vehicles traveling on *Road 54907* are mainly from the *Road 54705, 28930* and *30874*, with their *K-Nearest Adjacent Ratios* of 0.45, 0.30 and 0.18 respectively. This indicates that the vehicles on *Road 54907* mainly come from two paths: (1) 54 705 → 28 930 → 21 306 → 54 907, and (2) 54 705 → 30 874 → 28 918 → 43 059 → 54 907.

5.3.6. Effectiveness on other dataset

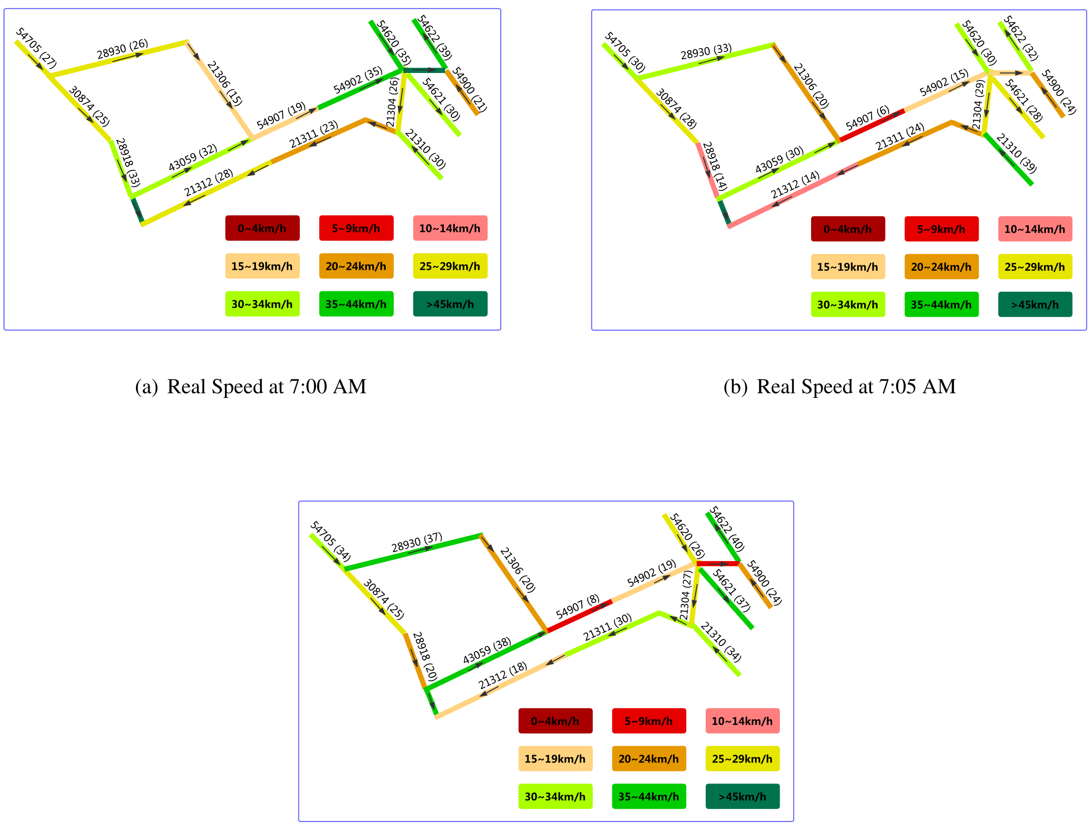
To confirm the robustness of NN_Forecast, we also compared it with others on the dataset from Hangzhou, another big city in eastern China. We investigated the real 97,158,000 GPS records of 8385 moving vehicles in Hangzhou, collected from June 5, 2012 to June 7, 2012, with 32,297,000 GPS records on first day, 32,399,000 GPS records on the second day, and 32,462,000 GPS records on the third day. As Table 7 indicates, our approach outperforms the other three when predicting the traffic speed in 5-min and 10-min, which is the same as the result based on the Wenzhou dataset.

6. Conclusion

To predict traveling speed in short-term is a key problem for real-time traffic control and trip planning. In this paper, we present a novel approach to short-term traffic speed prediction. Firstly, we determine the K most relevant adjacent roads for each road and assign their weights according to the influence degree. Afterwards,

Table 7
Comparison of speed prediction accuracy on the Hangzhou dataset.

Time interval (min)	Methods	≤2 km/h	≤3 km/h	≤4 km/h	≤5 km/h	≤10 km/h	MAPE	RMSE
5	NN_Forecast	20.48%	29.86%	38.73%	47.87%	75.69%	15%	8.96
	Improved KNN	16.22%	24.70%	32.47%	38.55%	64.11%	31%	15.47
	C-WSVM	17.94%	26.08%	34.01%	41.06%	66.27%	27%	14.11
	ARIMA	14.21%	21.60%	28.61%	35.313%	59.69%	33%	13.44
10	NN_Forecast	17.87%	25.47%	33.57%	40.78%	66.80%	22%	11.46
	Improved KNN	14.93%	22.27%	29.66%	36.17%	60.23%	34%	16.97
	C-WSVM	16.37%	23.76%	30.97%	37.87%	61.11%	29%	15.72
	ARIMA	15.88%	23.06%	30.69%	37.10%	62.40%	27%	12.68
15	NN_Forecast	13.62%	20.59%	26.59%	32.41%	55.72%	33%	15.66
	Improved KNN	11.47%	17.59%	23.47%	30.39%	51.22%	41%	18.67
	C-WSVM	14.59%	21.55%	28.36%	34.78%	57.08%	34%	16.31
	ARIMA	16.92%	24.39%	31.42%	38.06%	64.72%	24%	11.93



(a) Real Speed at 7:00 AM

(b) Real Speed at 7:05 AM

(c) Predicted Speed at 7:05 AM

Fig. 7. Impact of parameter nc on prediction accuracy (5-min interval).

we obtain the relationship between the future traffic speed of once certain road and the traffic attributes of its adjacent roads, and determine the *M* most relevant attributes for each road and the corresponding weights. In order to construct the piecewise function of traffic speed with traffic attributes, we employ an improved Jenks clustering method to obtain the optimal segmentation of traffic attribute values. Finally, we establish the prediction model for short-term speed based on most relevant adjacent roads and most relevant traffic attributes, and employ the Particle Swarm Optimization (PSO) algorithm to obtain the model parameters. We conducted the extensive experiment based on the real traffic data from Wenzhou. The experimental results demonstrated our approach has higher stability and accuracy, especially for 5-min and 10-min speed prediction, compared with the state-of-the-art approaches.

Our study shows that it is effective to use the multiple traffic attributes of adjacent roads to forecast the short-term traffic speed. In the future, we plan to improve our proposed approach in the following aspects. (1) In order to get more convincing conclusions, more floating vehicle data are required for the large scale of real case experiment. (2) The current version of our approach uses the fixed number of adjacent roads and traffic attributes for all roads. In facts, how many adjacent roads and their traffic attributes influence one certain road may vary. Therefore, we will consider different numbers of adjacent roads and traffic attributes for target roads so as to further improve the performance of our approach.

Acknowledgments

The work was supported by National Natural Science Foundation of China (No. 61100043), the Key Science and Technology Project of Zhejiang Province, China (No. 2017C01010). The work was also supported by CSIRO Data61, and ARC Discovery Projects DP140103499 and DP160102412, Australia.

References

[1] H. Zhang, Link-journey-speed model for arterial traffic, Transp. Res. Rec. J. Transp. Res. Board 1676 (1676) (1999) 109–115.
[2] S. Chawla, Y. Zheng, J. Hu, Inferring the root cause in road traffic anomalies, in: 2012 IEEE 12th International Conference on Data Mining, IEEE, 2012, pp. 141–150.
[3] J. Xu, T. Tang, J. Yan, Z. Liu, Short-term traffic flow forecasting based on multi-machine learning competitive strategy, J. Transp. Syst. Eng. Inf. Technol. 16 (4) (2016) 185–190.
[4] J. Myung, D.-K. Kim, S.-Y. Kho, C.-H. Park, Travel time prediction using k nearest neighbor method with combined data from vehicle detector system and automatic toll collection system, Transp. Res. Rec. J. Transp. Res. Board 20 (2256) (2011) 51–59.
[5] S. Clark, Traffic prediction using multivariate nonparametric regression, J. Transp. Eng. 129 (2) (2003) 161–168.
[6] B. Zhang, G. Trajcevski, F. Liu, Clustering speed in multi-lane traffic networks, in: 2016 ACM 25th International on Conference on Information and Knowledge Management, ACM, 2016, pp. 2045–2048.
[7] X.Y. Chen, H.K. Pao, Y.J. Lee, Efficient traffic speed forecasting based on massive heterogenous historical data, in: IEEE International Conference on Big Data, IEEE, 2015, pp. 10–17.
[8] S. Jeon, B. Hong, Monte carlo simulation-based traffic speed forecasting using historical big data, Future Gener. Comput. Syst. 65 (2016) 182–195.
[9] C. Van Hinsbergen, A. Hegyi, J. Van Lint, H. Van Zuylen, Bayesian neural networks for the prediction of stochastic travel times in urban networks, IET Intell. Transp. Syst. 5 (4) (2011) 259–265.
[10] X. Ma, Z. Tao, Y. Wang, H. Yu, Y. Wang, Long short-term memory neural network for traffic speed prediction using remote microwave sensor data, Transp. Res. C 54 (2015) 187–197.
[11] B. Jiang, Y. Fei, Traffic and vehicle speed prediction with neural network and hidden markov model in vehicular networks, in: Intelligent Vehicles Symposium (IV), 2015 IEEE, IEEE, 2015, pp. 1082–1087.
[12] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction, Sensors 17 (4) (2017) 818.
[13] J. Wang, Q. Gu, J. Wu, G. Liu, Z. Xiong, Traffic speed prediction and congestion source exploration: a deep learning method, in: IEEE International Conference on Data Mining, IEEE, 2016, pp. 499–508.
[14] N.G. Polson, V.O. Sokolov, Deep learning for short-term traffic flow prediction, Trans. Res. C 79 (2017) 1–17.
[15] Y. Jia, J. Wu, Y. Du, Traffic speed prediction using deep learning method, in: 2016 IEEE 19th International Conference on Intelligent Transportation Systems, IEEE, 2016, pp. 1217–1222.
[16] D. Kehagias, A. Salamanis, D. Tzovaras, Speed pattern recognition technique for short-term traffic forecasting based on traffic dynamics, IET Intell. Transp. Syst. 9 (6) (2015) 646–653.
[17] M.T. Asif, J. Dauwels, C.Y. Goh, A. Oran, E. Fathi, M. Xu, M.M. Dhanya, N. Mitrovic, P. Jaillet, Spatiotemporal patterns in large-scale traffic speed prediction, IEEE Trans. Intell. Transp. Syst. 15 (2) (2014) 794–804.
[18] V. Fabrizi, R. Ragona, A pattern matching approach to speed forecasting of traffic networks, Eur. Trans. Res. Rev. 6 (3) (2014) 333–342.
[19] J. Xia, M. Chen, W. Huang, A multistep corridor travel-time prediction method using presence-type vehicle detector data, J Intell. Transp. Syst. 15 (2) (2011) 104–113.
[20] E. Jenelius, H.N. Koutsopoulos, Travel time estimation for urban road networks using low frequency probe vehicle data, Transp. Res. B 53 (2013) 64–81.
[21] F. Zheng, H. Van Zuylen, Urban link travel time estimation based on sparse probe vehicle data, Transp. Res. C 31 (2013) 145–157.
[22] A. Simroth, H. Zahle, Travel time prediction using floating car data applied to logistics planning, IEEE Trans. Intell. Transp. Syst. 12 (1) (2011) 243–253.
[23] M.A. Rasyidi, J. Kim, K.R. Ryu, Short-term prediction of vehicle speed on main city roads using the k-nearest neighbor algorithm, J. Intell. Inf. Syst. 20 (1) (2014) 121–131.
[24] H. Hu, G. Li, Z. Bao, Y. Cui, J. Feng, Crowdsourcing-based real-time urban traffic speed estimation: from trends to speeds, in: 2016 IEEE 32nd International Conference on Data Engineering (ICDE), IEEE, 2016, pp. 883–894.
[25] P. Dell'Acqua, F. Bellotti, R. Berta, A. De Gloria, Time-aware multivariate nearest neighbor regression methods for traffic flow prediction, IEEE Trans. Intell. Transp. Syst. 16 (6) (2015) 3393–3402.
[26] W. Hu, L. Yan, K. Liu, H. Wang, A short-term traffic flow forecasting method based on the hybrid pso-svr, Neural Process. Lett. 43 (1) (2016) 155–172.
[27] B. Yao, C. Chen, Q. Cao, L. Jin, M. Zhang, H. Zhu, B. Yu, Short-Term traffic speed prediction for an urban corridor, Comput.-Aided Civ. Infrastruct. Eng. 32 (2) (2017) 154–169.
[28] J. Wang, Q. Shi, Short-term traffic speed forecasting hybrid model based on chaos–wavelet analysis-support vector machine theory, Transp. Res. C 27 (2) (2013) 219–232.
[29] A.R. YG, S. Kumar, H. Amaresh, H. Chirag, Real-time speed estimation of vehicles from uncalibrated view-independent traffic cameras, in: TENCON 2015-2015 IEEE Region 10 Conference, IEEE, 2015, pp. 1–6.
[30] G.-H. Ahn, Y.-K. Ki, E.-J. Kim, Real-time estimation of travel speed using urban traffic information system and filtering algorithm, IET Intell. Transp. Syst. 8 (2) (2013) 145–154.
[31] Z. He, B. Cao, Y. Liu, Accurate real-time traffic speed estimation using infrastructure-free vehicular networks, Int. J. Distrib. Sens. Netw. (2015) 6.
[32] M.-C. Tan, S.C. Wong, J.-M. Xu, Z.-R. Guan, P. Zhang, An aggregation approach to short-term traffic flow prediction, IEEE Trans. Intell. Transp. Syst. 10 (1) (2009) 60–69.
[33] B. Deng, S. Denman, V. Zachariadis, Y. Jin, Estimating traffic delays and network speeds from low-frequency gps taxis traces for urban transport modelling, EJTIR 15 (4) (2015) 639–661.
[34] J. Zhang, S. He, W. Wang, F. Zhan, Accuracy analysis of freeway traffic speed estimation based on the integration of cellular probe system and loop detectors, J. Intell. Transp. Syst. 19 (4) (2015) 411–426.
[35] T. Katsuki, T. Morimura, M. Inoue, Traffic velocity estimation from vehicle count sequences, IEEE Trans. Intell. Transp. Syst. (99) (2016) 1–13.
[36] J. Chen, S. Yang, H. Li, B. Zhang, J. Lv, Research on geographical environment unit division based on the method of natural breaks (jenks), Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XL-4 /W3 (4) (2013) 47–50.
[37] Y. Zhang, B. Shi, Non-tradable shares pricing and optimal default point based on hybrid kmv models: evidence from china, Knowl.-Based Syst. 110 (2016) 202–209.