Transportation Research Part C 108 (2019) 130–150

Contents lists available at ScienceDirect

Transportation Research Part C

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

A tailored machine learning approach for urban transport network flow estimation☆

Zhiyuan Liu[a,*], Yang Liu[a], Qiang Meng[b], Qixiu Cheng[a]

[a] Jiangsu Key Laboratory of Urban ITS, Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, School of Transportation, Southeast University, China

[b] Department of Civil and Environmental Engineering, National University of Singapore, Singapore

ARTICLE INFO

Keywords:
Transport network flow estimation
Random forest
Gradient boosting
Feature extraction
Cellphone location data
License plate recognition data

ABSTRACT

This study deals with urban transport network flow estimation based on Cellphone Location (CL) and License Plate Recognition (LPR) data. We first propose two methods to filter CL data and extract the spatio-temporal traffic features for a specific road segment. A tailored machine learning approach is developed, including two components: a tangible multi-grained scanning ensemble learning model and a novel two-stage zero-shot learner. The former aims to estimate traffic flow on a single link with both filtered CL data, extracted spatio-temporal traffic features, and LPR data by incorporating the unique merits thereof. The latter is capable of estimating traffic flow on those links with only CL data by considering the spatial features of these links and relevant land-use information. Finally, case studies are analysed to demonstrate the impressive performance of the tailored machine learning approach.

1. Introduction

Transport network flow estimation is of great importance in the network performance evaluation, either in a microscopic level for a particular road segment (Daganzo, 1997) or in a macroscopic level for the whole transport network (Sheffi, 1985). Existing methods to analyse the transport network flow are mostly in two ways, i.e., analytical approaches and simulation-based approaches.

In general, the analytical approaches include the conventional four-step model (Sheffi, 1985), the activity-based model (Lam and Yin, 2001), and the integrated transport-land use model (Hunt and Simmonds, 1993), *etc.*, As one of the most widely used analytical approaches, the classical four-step model (or combining some two or three steps in the four-step model) for urban transportation planning has been advocated for over 60 years (Cascetta, 2009). The four-step modelling approach has been also widely used in most transport planning software packages, such as EMME, TransCAD, CUBE, and VISUM. The four-step model necessitates the collection of various data such as land-use, zoning, household numbers, car ownership, *etc.*, With these collected input data, it yields the travel demand as well as link flows (de Dios Ortuzar and Willumsen, 2011). However, in the current analytical approaches, there are some strong assumptions that cannot well reflect the real circumstances, for instance, the Wardrop's principle of user equilibrium in travelers' route choice behaviours assumes that all the travelers choose the shortest path and are never cooperative, which is not always true in practice. These assumptions/limitations thus undermine the accuracy of estimated/predicted flows by the analytical approaches.

The simulation-based approaches rely on the specific definition of the reaction rules between vehicle-vehicle (car-following, lane

☆ This article belongs to the Virtual Special Issue on "Machine learning".

* Corresponding author.
E-mail address: zhiyuanl@seu.edu.cn (Z. Liu).

https://doi.org/10.1016/j.trc.2019.09.006
Received 6 December 2018; Received in revised form 29 August 2019; Accepted 7 September 2019
Available online 24 September 2019
0968-090X/

changing, gap acceptance) as well as between vehicle-environment (Panwai and Dia, 2005). These reaction rules are usually assumed to follow a certain random distribution and implemented with pseudo-random numbers. However, it is also very challenging to accurately obtain these distributions that well reflect the practice, which accordingly affects the predicted flows and results.

In recent years, benefiting from the rapid development of the intelligent transport system facilities as well as the algorithms in data analytics, the massive data collected in urban areas has provided new possibilities to develop another pathway for the analysis of transport network flows. Herein, the new pathway is to use the data-driven computational learning theory in artificial intelligence (Jordan and Mitchell, 2015) for transport network flow analysis, which is addressed in this paper.

For the estimation of transport network flows, the relevant data include sensor-based data (*e.g.*, license plate recognition (LPR) data, loop detector data, and RFID (Radio Frequency Identification Device) data), vehicle GPS data, cellphone location (CL) data, metro and IC (Integrated Circuit) card transaction data, among others (Ban et al., 2011; Gao and Liu, 2013; Chen et al., 2016). Amongst these data types, CL data have high market penetration and spatio-temporal coverage. The CL data also get frequent real-time dynamic updates and incur low collection cost, and thus it is suitable for the analysis of total traffic volume in the entire network. However, its positioning technique has an error of around 100 m (Ho, 2013), and such an error makes it difficult to locate the phone users to particular links, especially in dense urban areas.

Other sensor-based data (*e.g.*, LPR data) have a high positioning accuracy (Zhan et al., 2015; Mo et al., 2017), and can be regarded as an accurate reflection of the traffic volume. Nevertheless, sensor devices are only installed on certain locations with low spatial coverage. Thus, only one data source cannot possess the high positioning accuracy and high coverage simultaneously, which are both needed for flow estimation. It is necessary to develop a technique that combines the merits of the two aforementioned data. In this paper, we built a machine learning approach, which is completely data-driven without theoretical assumptions (*e.g.*, the arrival of vehicles follows a time-dependent Poisson process), to estimate the traffic flow using the selected LPR and CL data.

1.1. Literature review

Computational learning theory stems from Artificial Intelligence (AI) devoted to studying the design and analysis of machine learning algorithms (Jordan & Mitchell, 2015). It is crucial to choose an appropriate machine learning algorithm that can learn from and make estimations on the collected data (Han and Kember, 2000; Sun et al. 2015; Liu et al., 2018). Machine learning approaches can be classified into three categories: (i) supervised learning, (ii) unsupervised learning and (iii) reinforcement learning (Alpaydin, 2014). Most studies on road traffic flows using big data focus on flow predictions (Chen, 2017; Li et al., 2018; Liu et al., 2019; Mehran and Kuwahara, 2013). The review in this section focuses on the machine learning methods for urban traffic flow analysis.

We first review the supervised learning methods. Sun et al. (2006) proposed a spatio-temporal Bayesian network prediction approach, trying to leverage the information that abounds in a transportation network. To deal with the frequent fluctuations and abrupt changes therein, Chang et al. (2012) proposed a dynamic multi-interval method based on the KNN regression for traffic flow prediction. Wei et al. (2012) presented a hybrid forecasting model, where empirical mode decomposition and artificial neural networks are combined, to predict short-term passenger flow in subway systems. Jeong et al. (2013) proposed an on-line learning weighted support-vector regression (SVR) prediction model for short-term traffic flow forecasting. Lv et al. (2015) developed a stacked autoencoder (SAE) model for the traffic flow prediction, considering non-linear spatio-temporal correlations in traffic data. The studies of Ban et al. (2018) and Wang et al. (2019) handled the multi-source data using a Divide-Conquer-Integrate framework, and the power of fusing data from multiple sensors in trip estimation was extensively revealed.

As an enhanced supervised learning model, ensemble learning has been conducted to combine several weak base learners (Chen et al., 2017; Liu et al., 2019b). As to the deep learning models, Huang et al. (2014) proposed a deep learning framework that is constituted of a deep belief network (DBN) and a multi-task regression layer to forecast short-term traffic flow. Ma et al. (2015) utilised RNN for traffic speed prediction using microwave sensor data. Zhang et al. (2016) partitioned a city into a grid map and presented a novel spatio-temporal residual network to forecast crowd flows in each region. Traffic flow data have non-linear features due to frequent changes between free flow, congestion, breakdown, and resumed situations. To tackle these challenges, Polson et al. (2017) proposed the use of a deep learning architecture.

Unsupervised learning methods are also used in traffic or passenger flow predictions for the cases with no label (true values), which can discover hidden patterns in the raw data. The most typical applications of unsupervised learning are clustering (*e.g.*, *k*-means) and dimensionality reduction analysis (*e.g.*, principal component analysis, PCA). In many studies, unsupervised learning is often used to reduce the dimensionality. Jin et al. (2007) proposed a multi-dimensional method based on PCA-SVR for forecasting short-term network traffic flow simultaneously. Based on PCA, Djukic et al. (2012) explored the potential for dimensionality reduction in the analysis of Origin-Destination (OD) demand, which is helpful in improving the quality of OD estimates with the standard Kalman filter approach. Integrating CL data and sensor data, Wu et al. (2015) proposed an optimisation method based on the projected gradient algorithm to calculate path flows in urban road networks. Wu et al. (2018) developed a layered computational graph that integrates four types of data and calculates the travel demand, path flows, and link flows in an urban road network. Sekula et al. (2018) used taxi probe data and sensor data to estimate the hourly traffic volumes using neural network models.

We can see that fusion of sensor data with other types of widely-spread data are needed (cellphone data or probe vehicle data) for the estimation of network-wide traffic flows. However, when sensor data (used as ground-truthing markers or as a label for traffic flows) are transferred for the analysis of other links with no label, it is *per se* a transfer learning problem (Wang et al., 2011; Duan et al., 2012). Transfer learning methods are used when a target domain is lack of training data, and it transfers information from a

relevant domain with sufficient data. This study thus aims to employ the transfer learning method to the urban traffic flow estimation problem.

1.2. Objectives and con tributions

This study aims to develop a tailored machine learning approach using multi-source big traffic data for urban transport network flow estimation, in that it tackles the following two difficulties: first, on those links with both CL and LPR data, how to develop a machine learning model that can estimate traffic volumes only taking CL data as inputs; second, for network-wide flow estimation, actual traffic flows for most road segments are unavailable due to the limited number of LPR devices and the varied flow distribution between different segments, resulting in a zero-shot learning problem. To solve the zero-shot learning problem, more additional features, including spatial features, regional function features, and their intrinsic characteristics, are required and extracted. Finally, a zero-shot transfer learning transport network flow estimation model, where all the links with actual traffic flows are taken as a pool, is used to build a model for flow estimation. To the best of our knowledge, this study makes the first attempt for the transport network flow estimation problem using transfer learning.

To sum up, the contributions of this study are two-fold: firstly, on each link with both CL and LPR data, a traffic volume estimation model is proposed, which can be used to estimate the flow on this link using only CL data. The spatio-temporal characteristics of CL data are taken into full consideration. Furthermore, we improve the accuracy by adding multi-grained features and using the sliding window method. Secondly, combined with the characteristics of the transport network, a network-wide traffic flow estimation model for urban areas is developed. The zero-shot learning problem is solved, and the spatial features, the regional function, and the intrinsic characteristic of the road are also considered.

It should be noted that this paper aims to estimate the 100% traffic volume in the entire network (complete spatial and temporal coverage). With the aids of existing communication and detection technologies, it is not difficult to get a portion of traffic volume (less than 100%), nor to get the 100% volume in only one location (small spatial coverage), but very challenging to get the 100% volume wherever and whenever. The usage of multi-source data (CL data and LPR data) maintain the advantage of high coverage and high accuracy altogether. In addition, our method no longer needs predefined parameters (e.g., setting up a threshold) in naïve methods. Instead, supervised learning is employed, which is completely data-driven and free of assumptions.

The remainder of this paper is organised as follows: Section 2 presents the research problem. Section 3 introduces the tree-based supervised machine learning model, Section 4 presents solutions to the traffic flow estimation problem for a single road segment, while the network-wide flow estimation for urban areas is analysed in Section 5. Section 6 provides two case studies to validate the performance of the proposed tailored machine learning approach, and Section 7 presents the conclusions.

2. Data description and problem statement

2.1. Characteristics of the input data

Consider an urban transportation network denoted by $G = (N, A)$, where N is the set of nodes and A is the set of links. Let $\bar{A} \subseteq A$ denote the subset of links with LPR facilities/data and $A^{'} \subseteq A$ denote the subset of links without LPR facility/data. The CL data collected by one telecommunication company are used; herein CL data refer to triangulation data (Ho, 2013).

The CL data have high coverage, receive real-time dynamic updates, and incur low collection cost. For example, the market penetration of a telecommunication company can reach up to 70% in a mega-city, and such a sampling rate is much higher than that of many other traffic data types, *e.g.*, floating vehicles, taxi probes, loop detector data, *etc*. Note that the CL data adopted in this study are all 4G data. Table 1 lists CL data tags.

CL data suffer from three major disadvantages:

(a). It is difficult to identify the travel mode of each phone user. For example, the travel speed estimated cannot reasonably differentiate between walking, cycling, driving, or public transport modes. This is because the stopping of vehicles at intersections makes it difficult to measure real travel speeds. Moreover, it is difficult to recognise ride-sharing passengers, making it challenging to convert a certain number of moving cellphones into an accurate traffic volume.

(b). CL data have a location error of about 100 m (Ho, 2013) making it difficult to separate travellers from residents in adjacent buildings areas. In addition, within a 100-m range, there could be more than one road link, and it is difficult to associate a cellphone sample to a specific link.

(c). The temporal resolution of CL data is not fixed, *e.g.*, the sampling interval of each cellphone varies constantly. If users

Table 1
CL data tags.

User ID	Access time	Latitude	Longitude
460110120767844	2016/10/19 17:33:25	31.96077	118.797097

Table 2
LPR data lags.

Vehicle ID	Access time	Vehicle type	Latitude	Longitude	Lane number	Spot speed
1261001772	2016/10/19 11:03:17	3	31.978	118.764	1	32.2

communicate with the base station at a fixed time interval (*e.g.*, every 30 s), the vehicle can be identified by a users' trajectory. For example, if multiple user trajectories are identical over a long period of time, at a higher speed, then it can be concluded that they are in the same vehicle. However, the data record gaps and frequency are highly heterogeneous for different phone users, making it challenging to identify ride-sharing passengers.

These major disadvantages make it hard to estimate urban traffic flows using only CL data. Due to these limitations, current studies using the CL data mainly focus on the freeway traffic state analysis (Herrera et al., 2010), OD demand forecasting (Alexander et al., 2015; Iqbal et al., 2014), and traffic zoning (Dong et al., 2015).

Accurate sensor-based data are needed to fuse with the CL data. The LRP data are thus taken by this study, which collect the traffic volume on all lanes at a point (Table 2).

The LPR data allow high recognition accuracy. For example, the LPR data in China have at least 95% recognition accuracy in the daytime and 90% at night for the detection of vehicles with number plates according to Chinese national standards (Ministry of Public Security, 2016). In other words, the LPR data can be regarded as an accurate approximation of actual traffic flows. Although LPR data can accurately reflect the traffic flow on an installation site, its coverage is very low due to the limited number of LPR sites/facilities. It should be noted that the LPR system used in this study is mounted on gantries at road segments (between intersections) for traffic speed and volume detection, which measures and collects traffic volumes from all lanes, and thus are sufficient for model development.

The data used for the demonstration are from Nanjing China, which are also taken as the case study in Section 6. As shown in Fig. 1, the study area contains 36 bidirectional road links. Fig. 1 also shows the sites of LPR facilities (the red pins), and only 24 out of the 36 links are mounted with LPR facilities. For the 24 LPR sites in the study area, the total number of data records each day is about 130,000.

Fig. 1. The study area and LPR sites.

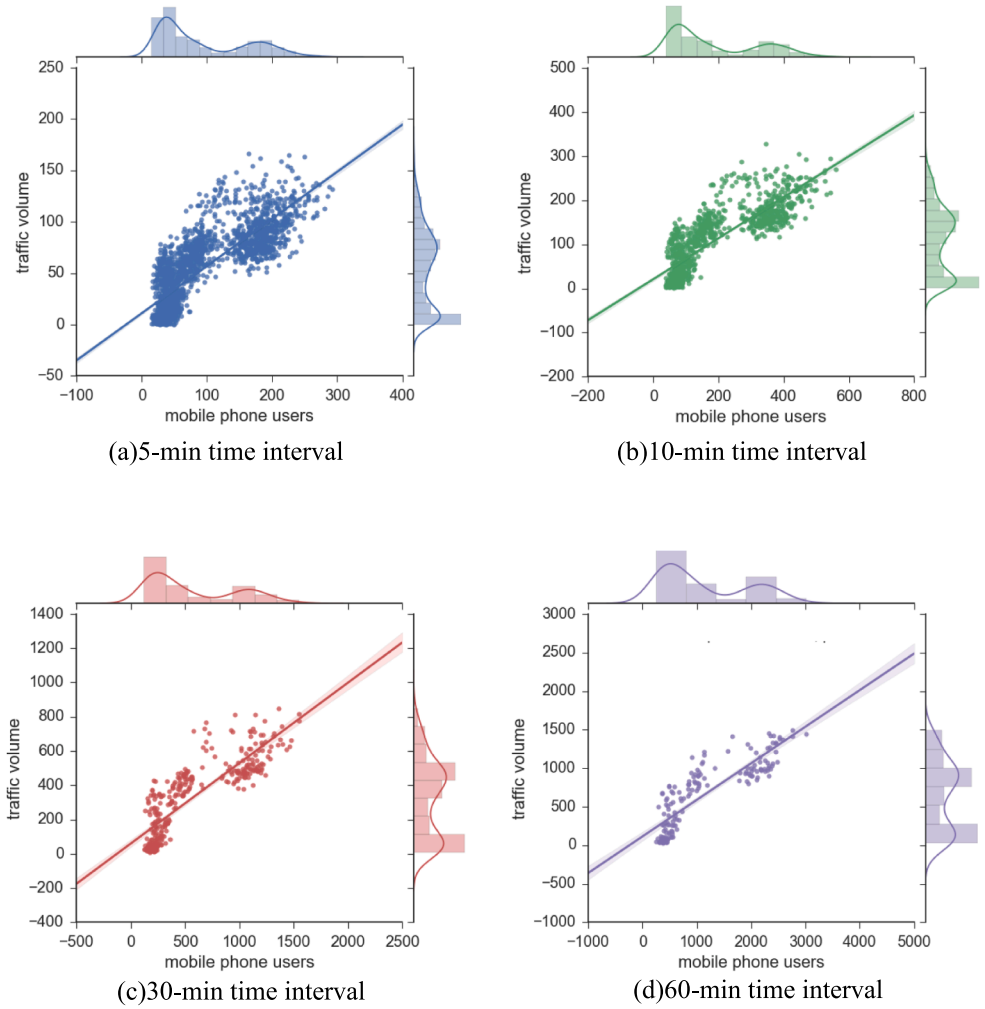


Fig. 2. The scatter diagram and histogram of CL data with the LPR data.

The 100-m positioning error of CL data contain two types of troublesome data: ping-pong data and drift data. Considering a cellphone whose location is close to the boundary of two base stations, ping-pong data may be generated because of frequently switching between these two stations. Moreover, if a cellphone user is fast-moving and switching between stations, drift data might also be generated. To avoid interference from ping-pong and drift data, they have been filtered from the dataset using the cleaning approach proposed by Yang et al. (2014).

2.2. Correlation test

To first verify the rationale of integrating these two types of data, a correlation test is conducted based on the Pearson correlation coefficient (PCC). The PCC is widely used in measuring similarity in machine learning (Chen, 2014), to quantify the correlation between two types of data:

$$PCC = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

where n is the number of samples, (x_i, y_i) reflects location of the i-th sample; $\bar{x}$ and $\bar{y}$ is the mean of x_i, $i = 1, 2, ...,n$ and y_i, $i = 1, 2, ...,n$, respectively.

As shown in Fig. 2, for the correlation test, we have run four scenarios, where the sample data are generated from the raw data (continuous CL and LPR data) in 5-min, 10-min, 30-min, and 60-min intervals, respectively. In Fig. 2, the data for each scenario are presented with their scatter diagram and histogram. Each scenario shows a good linear correlation, with PCC values of 0.82, 0.83,

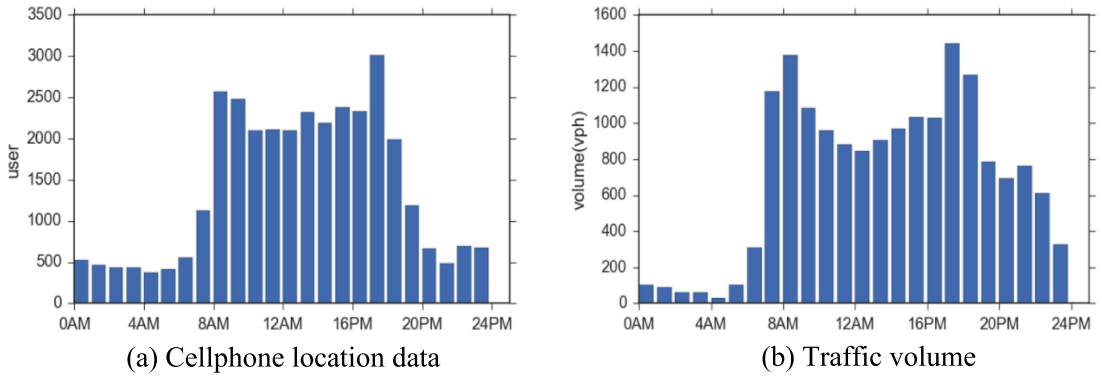


Fig. 3. Temporal distribution of mobile phone users and traffic volume in a segment of Ruanjian Road.

0.84, and 0.86 respectively. The clear linear correlation shown in Fig. 2 underpins the rationality of combining CL and LRP data, as a circumstantial evidence.

The links in set A contain both CL and LPR data, and each of these links is taken for subsequent data fusion. Without loss of generality, Ruanjian Road (the vertical link marked by tag 6105 in Fig. 1) is taken as an example to illustrate the methodology. Taking 60-min as the interval for aggregation, Fig. 3 shows the temporal variation of cellphone users and traffic volumes (LPR data) in Ruanjian Road, with morning and evening peaks. The traffic volume cannot be obtained by simply switching the CL data with a coefficient. The largest difference between these two types of data is in the time intervals between 0am to 4am, where the traffic volume is trivial yet the amount of CL data is much higher. Herein, the CL data are mainly generated by active phone users (*e.g.*, heartbeat packets of Apps) in adjacent residential areas, which are taken as a disturbance to the analysis of traffic volumes and are difficult to remove. To remedy these problems with CL data, we need to extract more features relating to the road links.

The CL data widely spread across the entire network yet only some road links are equipped with LPR facilities. Hence, we aim to cope with the following two issues in traffic flow estimation by fusing CL and LPR data.

Issue 1 – single link model: for any link $a \in A$ with LPR facilities, how can we develop a supervised machine learning model, denoted by H^a, to learn a function from CL data to traffic volumes (*i.e.*, LPR data)?

Issue 2 – network model: based on the developed supervised machine learning models H^a, how can we estimate traffic flow patterns on other links in A' where only CL data are available?

Note that for **Issue 1**, once the LPR equipment in link $a \in A$ is closed or broken, we can use the trained learner H^a to estimate traffic flow on this road segment with only CL data. We aim to build a tailored machine learning approach that can elegantly cope with these two issues. Before describing the specific models for the two issues, the methods used for data filtering and feature extraction are first discussed in Section 3.

3. CL data filtering and spatio-temporal feature extraction

3.1. A fine-grained method for CL data filtering

Considering that the CL data has a positioning error of 100 m, and that phone users in surrounding residential areas keep

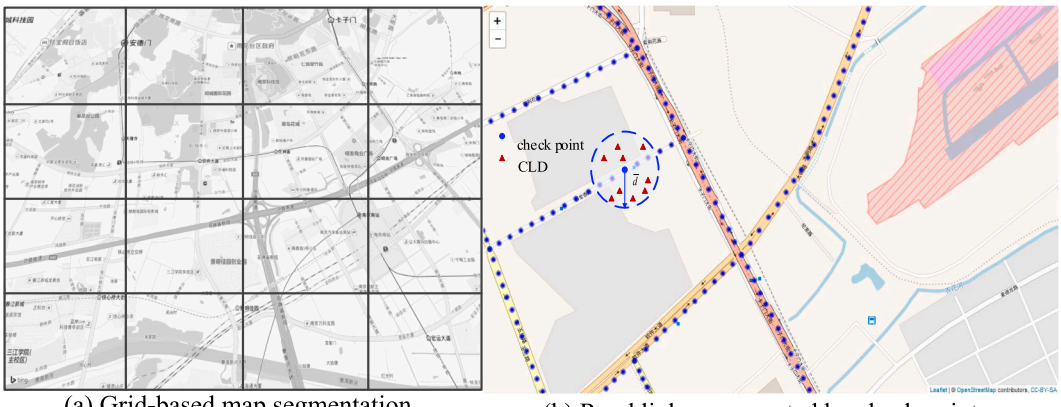


Fig. 4. Division of the study area.

generating data records, CL data are widely distributed across the whole study area. Hence, a method is first needed to filter the raw CL data and leave only those needed for road traffic flow estimation.

Existing studies using spatio-temporal data usually divide the whole study area into girds, as shown in Fig. 4(a), and then load the data into each grid, however, as mentioned in Section 2.1, a drawback of using CL data for traffic flow estimation is that it is affected by data from adjacent residents. Such a grid-based map segmentation method cannot surmount this drawback. We developed a fine-grained method for the filtering of CL data. As indicated in Fig. 4(b), on a two-dimensional e-map of the study area, each link is represented by a set of high-density check points. These can be considered as virtual sensors in each link to monitor traffic flows.

All the check points in link a are grouped into set P_a, and the coordinate of a check point $p \in P_a$ is denoted by x_p and y_p. With the aid of these check points, all CL records are marked, indicating whether they belong to a certain link a. We let CLD_i denote any CL data point i (its latitude and longitude are denoted by lat_i and lng_i). The distance $d_{i,p}$ between the location of CLD_i and a check point $p \in P_a$ equals:

$$d_{i,p} = 2 \cdot \bar{R} \cdot \arcsin\sqrt{\sin^2(\frac{\pi}{180}\Delta lat) + \cos(\frac{\pi}{180}lat_i)\cos(\frac{\pi}{180}x_p)\sin^2(\frac{\pi}{180}\Delta lng)} \tag{2}$$

where $\Delta lat = \frac{lat_i - x_p}{2}$, $\Delta lng = \frac{lng_i - y_p}{2}$ and $\bar{R}$ is the radius of the Earth (6371 km).

Based on Eq. (2), we can get the nearest check point for CLD_i, denoted by $p^*(i)$, as well as the distance between them, denoted by $d_{i,p^*(i)}$. Then, all CL data could be filtered based on this value $d_{i,p^*(i)}$. We use a threshold $\bar{d}$ (*e.g.*, $\bar{d} = 100$ m) to filter the CL data, which results in the following set S_a for each link $a \in A$, i.e., S_a: $=\{CLD_i | d_{i,p^*(i)} \leqslant \bar{d},\ p^*(i) \in P_a\}$. Using such a fine-grained data filtering method, it is convenient to analyse the speed and trajectories of each phone user, which reduces the influences from nearby residential phone users.

3.2. A tangible method to extract spatio-temporal features

Feature extraction/engineering is an essential step for the machine learning models, which largely affect the model accuracy. The features (also input data) determine the theoretical upper-bound performance of machine learning models overall, while specific models and algorithms can only try to approach such an upper limit. Therefore, rational selection and combination of features is a key contribution of the machine learning approaches developed in this paper.

It is still challenging to get features with strong generalisation ability and robustness in machine learning models. Thus, feature extraction should be problem-sensitive, where suitable domain knowledge should be accounted. CL data are typical spatio-temporal data, thus when designing features, its spatio-temporal characteristics should be considered.

Speed is a good reflection of the spatial features: for each link $a \in A$, we first calculate the travel speed of each phone user based on the filtered CL data in set S_a. With the provided user ID, all the CL data records for the same user could be identified. We let ($lng_{u,i}$, $lat_{u,i}$, $t_{u,i}$),($lng_{u,i+1}$, $lat_{u,i+1}$, $t_{u,i+1}$) denote the two consecutive CL data records j and $j+1$ of a user u, and then their spot speed $v_{u,j}$ is calculated as follows:

$$v_{u,j} = d_{u,j}/(t_{u,j+1} - t_{u,j})$$

where

$$d_{u,j} = 2 \cdot R \cdot \arcsin\sqrt{\sin^2(\frac{\pi}{180}\Delta lat_{u,j}) + \cos(\frac{\pi}{180}lat_{u,j+1})\cos(\frac{\pi}{180}x_{u,j})\sin^2(\frac{\pi}{180}\Delta lng_{u,j})}$$

herein, $\Delta lat_{u,j} = (lat_{u,j+1} - lat_{u,j})/2$, $\Delta lng_{u,j} = (lng_{u,j+1} - lng_{u,j})/2$.

Consequently, the average speed of each user at S_a can be calculated. This average speed is further quantified into L intervals (*e.g.*, six intervals: 0–10 km/h, 10–20 km/h, 20–30 km/h, 30–40 km/h, 40–50 km/h, and > 50 km/h). The number of users in each speed interval is calculated for every T minutes (*e.g.*, 1 min). The spatial features of CL data are tabulated in Table 3.

Apart from travel speed, travel behaviours vary by time of day, which should also be treated as a feature of the data (we therefore add the temporal features including time slice ID, hour of day, and the day of week, *etc.*)

4. Multi-grained scanning ensemble learning model for single link flow estimation

The **Issue 1** raised in Section 2 is addressed here: as claimed in Section 2, it is difficult to identify vehicles directly using CL data. Considering the good correlation and spatio-temporal matching of CL and LPR data, a supervised machine learning approach is ideal

Table 3
Spatial features of CL data in link a

Distance threshold	Features
$\bar{d}$	Total user in a time slice extracted fromS_a
	Number of users in speed interval [0, 10) km/h
	Number of users in speed interval [10, 20) km/h
	–

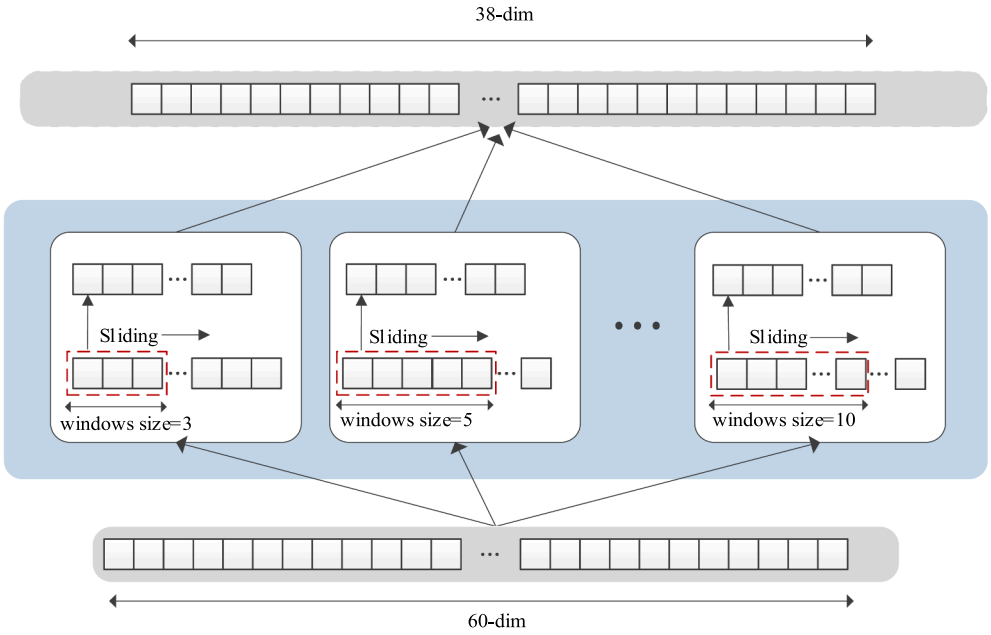


Fig. 5. The multi-grained scanning method.

for these two issues (Kotsiantis, 2007). In a supervised learning model, we have inputs X and the corresponding outputs Y, and the rule is to use an algorithm to learn the mapping/function $f(\cdot)$ from the inputs to the outputs, *i.e.*:

$$Y = f(X) \tag{3}$$

The goal is to learn a universal rule mapping the input values to the output values, *i.e.*, when we have new input data $\hat{X}$, we can estimate the output $\hat{Y}$ from that data. In this study, the input variables X are the features extracted from the CL data, while the output variable Y is the traffic flow in the corresponding road segment obtained from the LPR data. The traffic volume is analysed at an aggregate level, where all the travel modes are integrated for the analysis. The passenger car unit (pcu, as in Highway Capacity Manual 2010) is used to convert the multi-vehicle types to a unitary value. This aggregated traffic volume is taken as the label value Y in the supervised learning model $Y = f(X)$, where the CL data are the input X. Thus, although CL data cannot distinguish between different travel modes, the trained model can eventually output a reliable total traffic volume.

As mentioned above, the existing traffic volume estimation approach requires that we aggregate raw data using a fixed-size window. Such an aggregation reduces the sample sizes; thus the sliding window method is used, which considerably increased the sample size (see Appendix A).

4.1. A multi-grained scanning method

Data aggregation would also lose much raw information, especially when the time interval is large. To cope with this problem, we are inspired by image classification problems, where the feature map in the same layer use multiple different-size convolution kernel to obtain features at different scales. Combining these features, the classification effects are often better than using a single convolution kernel (Szegedy et al., 2015). Therefore, as a tailored approach for network flow estimation, we propose a multi-grained scanning method to extract different scales of information from raw data as new features for analysis.

As illustrated in Fig. 5, we treat the results obtained with a unified sliding window in Appendix A as intermediate results. Then, the multi-grained scanning method uses different sizes of window to scan the intermediate results in order, which gives rise to new features. For example, assuming that the intermediate results have 60 values, we can use three window sizes (3, 5, and 10) and the sliding step length matches the window size (Fig. 5). Thus, a sliding window of size 3 produces 20 features after scanning intermediate results, and the three different window sizes produce 20 + 12 + 6 = 38 features.

The multi-grained scanning method generates more data and also obtains new features at different scales, which can help to improve the accuracy of the estimated results.

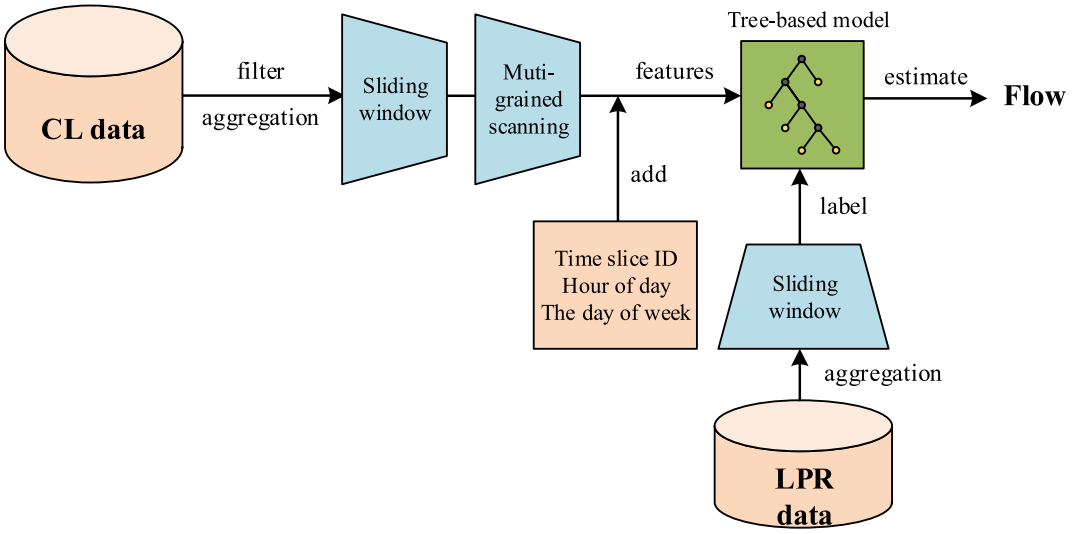


Fig. 6. Flowchart of tree-based multi-grained scanning ensemble learning model.

4.2. The multi-grained scanning ensemble learning model

Based on the techniques introduced above, we propose a tree-based multi-grained scanning ensemble learning model (a random forest model), and the structure of this model is introduced in Fig. 6. For each link $a \in A$, a tree-based model is trained, taking CL data as input and LPR data as labels. Eventually, $|A|$ models could be obtained by this supervised machine learning method.

5. Two-stage zero-shot learner for network-wide traffic flow estimation

This section copes with **Issue 2** raised in Section 2, where the trained $|A|$ learners based on links in A are used to estimate the flow on other links in A' with no LPR facility or data. Herein, the links in A are *source links* and those in A' are *target links*. A hurdle here is to get the suitable source link for each target link. The number of source links is limited, and the impacts of factors (*e.g.*, time, location, and weather) on their traffic flows are different. Thus, it is almost impossible to find a source link which is completely the same as a target link; therefore for any target link, all the $|A|$ source links are taken *as a pool* from which to build a new model for flow estimation. This method is, in nature, a zero-shot transfer learning transport network flow estimation model (see Section 5.2).

For the model in Section 5, the input data are: the CL and LPR data on the source links and CL data on the target links. The outputs are the estimated traffic volume on the target links.

5.1. Feature extraction for transport network flow estimation

It is more complicated to estimate the traffic flow for the network-wide urban areas than a single link (*i.e.*, **Issue 1** raised in Section 2) in the aspect of feature extraction. This is because spatial features (*e.g.*, the traffic flow pattern of the adjacent links may be similar, links belong to the same arterial road may have similar traffic flow characteristics, *etc.*), the features for regional land-use functions (*e.g.*, distributions of CL data on links in business and residential areas are clearly different), and the intrinsic characteristics (*e.g.*, tidal flow along some road segments) should be taken into consideration in network flow estimation in urban areas. They are elaborated as follows.

5.1.1. The spatial features of the road

A simple toy-size example (Fig. 7) is created to illustrate this point, where link 1 is a target link, while links 2 and 3 are two source

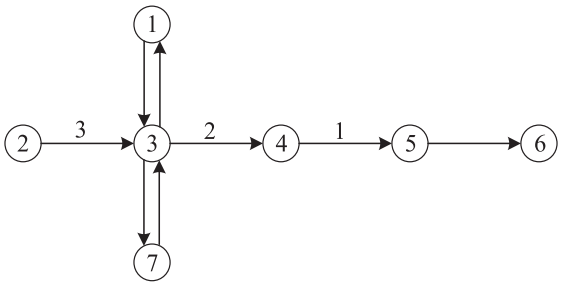


Fig. 7. A toy-size network example.

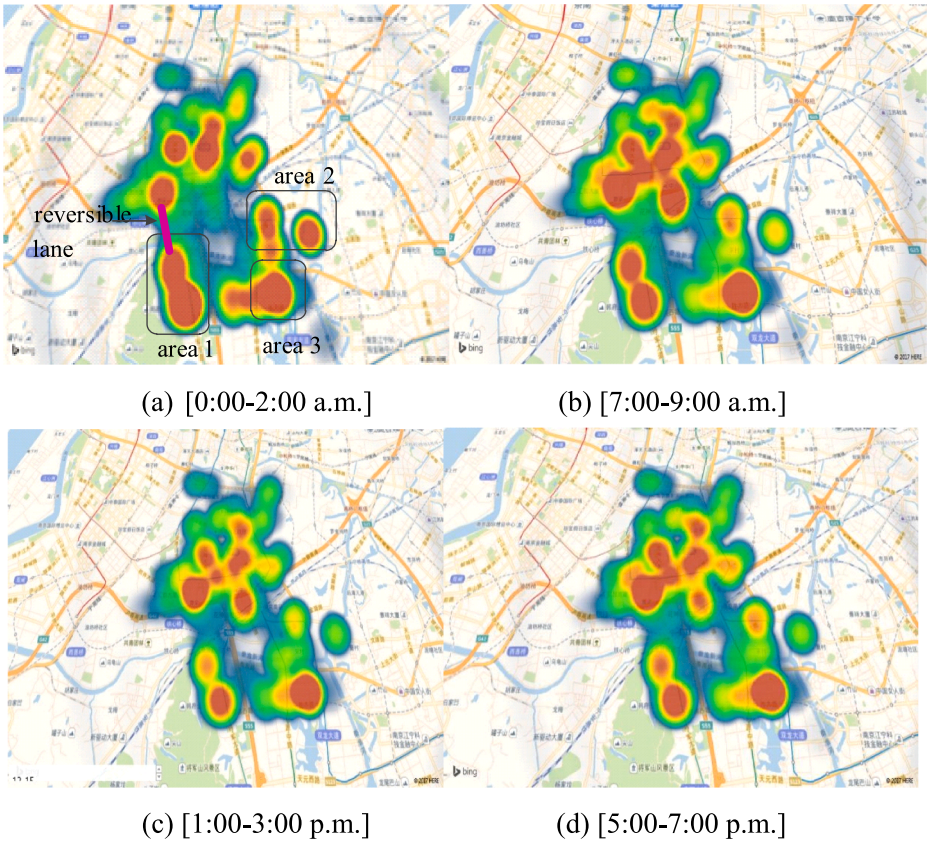


Fig. 8. Spatio-temporal distribution of cellphone location data.

links. The path from node 2 to node 6 is an arterial road, which contains a high traffic volume compared with the minor roads between nodes 1 and 7.

Adjacent links may have similar traffic flow characteristics (*e.g.*, flow on link 1 would be similar to that on link 2). In this regard, we take the tail and head nodes of each link as features, so any two links with the same node feature are spatially adjacent. As shown in Fig. 7, the flow on target link 1 is also related to that on a distant link in the same path (*i.e.*, link 3). Therefore, the head and tail nodes of the path (*e.g.*, path: $2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$) also serve as features for links in this path; *e.g.*, nodes 2 and 6 for links 1 to 3.

5.1.2. Features relating to land-use

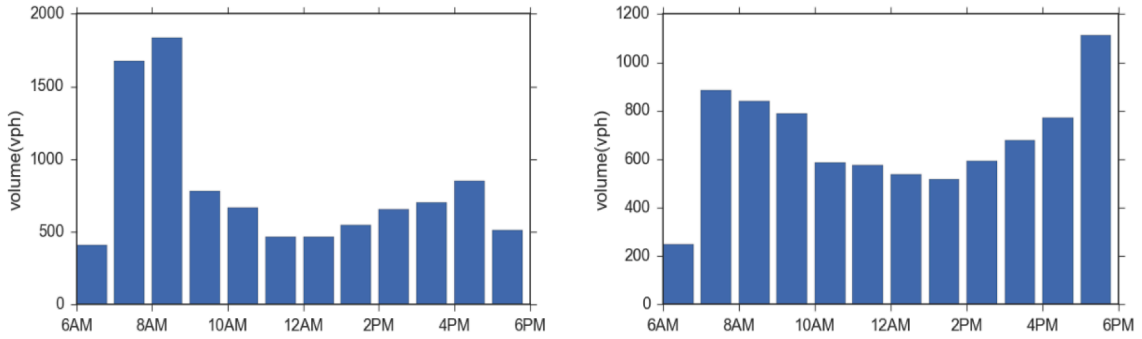
The spatio-temporal distribution of traffic volume in urban areas is influenced by its land-use function. Fig. 8 illustrates the spatio-temporal distribution of CL data in our study area at four representative time periods. As shown in Fig. 8(a), three areas are taken for illustration: the changing patterns of the CL data in these three areas are evident. In areas 1 and 2, the number of users decreases during the morning rush hours (7–9 a.m.) (users leave these areas continuously); while during the evening rush hours (5–7p.m.), the number of users rises again (users flow back to the area). During 1 to 3p.m., area 1 and 2 have the least CL data density among the four time periods; however, in area 3, the CL data density changes little over the 24 h, because area 3 is a residential-business mixed area, but areas 1 and 2 are residential areas.

Therefore, the land-use function has inherent impacts on the changes in traffic volume, which should be taken as features for the machine-learning model. In this study, the land-use data are obtained based on the on-line Gaode Map website. Here, we take three land-use types C = {Residential District, Commercial District, Other}. For each link, the land-use types on both sides are indicated.

5.1.3. Intrinsic characteristic of the link

We should further consider the intrinsic characteristic of each link (*e.g.*, the tidal phenomenon/nature of flow along some links). Here, we mainly consider two intrinsic characteristics: the number of lanes, and the link direction.

The data visualisation helps us to identify tidal effects accurately: for instance, area 1 in Fig. 8(a) has a continuous outflow during the morning rush hours and a continuous inflow during the evening rush hours (*i.e.*, Huashen Avenue, as shown in Fig. 9); therefore, a tidal effect arises in area 1.



(a) From south to north (b) From north to south

Fig. 9. Link flow on Huashen Avenue.

Based on the above analysis, there would be a clear tidal effect in the road connecting a busy district and a quiet district, so the two directions (to the hot district or leave the hot district) of the link are denoted as 0 and 1.

To sum up, these characteristics of the links in the network are denoted as $feature_{net}$, which includes the spatial features of the road, land use types, link directions, and the number of lanes. We use one-hot encoding to transform $feature_{net}$ into binary vectors. Since the order of magnitudes of CL data varied in different links (*e.g.*, 100 mobile users pass through link A per hour, while 1000 pass through link B per hour), $feature_{st}$ (*i.e.*, spatio-temporal features in Section 3.2) of each link should be standardised by the *z*-score method (see Section 5.2.1) to the same scale. We then merge $feature_{st}$ and $feature_{net}$ to train a zero-shot learner for transport network flow estimation.

All features used for the network flow estimation model are provided in Appendix B.

5.2. Zero-shot learner for transport network flow estimation

As mentioned, we cannot simply assume that the flow distribution of any link $a \in A'$ is similar to that of $a \in A$. Namely, there is no data for the analysis of links $a \in A'$, such a problem is termed as zero-shot learning problem in the area of transfer learning (Palatucci et al., 2009). Palatucci et al. (2009) extrapolated new image classes in the case of zero-data, where auxiliary information, in the form of a semantic knowledge base is added. The semantic knowledge base contains many sophisticated descriptions of the image properties. In general, a zero-shot learning problem is challenging, to the best of our knowledge, this is the first attempt to solve the transport network flow estimation problem with transfer learning.

Inspired by Palatucci et al. (2009) and their work on transfer learning, we presented a two-stage zero-shot learner for transport network flow estimation. Herein, we first build a traffic flow knowledge base, to get additional information for the zero-data transport network flow estimation problem. Let K denote the traffic flow knowledge base, which is a collection of {link ID, link capacity} data.

Let m denote the total number of input features hereafter, and these features yield an m-dimensional input space, denoted by X^m. The zero-shot learner first maps from X^m to a scaling label $\bar{Y}$ which is an intermediate result; and then maps this intermediate result to the final estimated value Y. The proposed two-stage process is reflected by two functions $\mathrm{F}(\cdot)$ and $\mathrm{L}(\cdot)$, respectively, a composite of which gives rise to the model $\mathrm{H}(\cdot)$:

$$\mathrm{H}(\cdot) = \mathrm{L}(\mathrm{F}(\cdot)) \tag{4}$$

where

$$\mathrm{F}: X^m \rightarrow \bar{Y} \tag{5}$$

$$\mathrm{L}: \bar{Y} \rightarrow Y \tag{6}$$

We use superscript s and t to indicate the source and target links, respectively. From the $|A|$ source links, we obtain N samples to build a training set D^s. Let (x_i^s, y_i^s) denote the i-th sample in D^s, where x_i^s is a m-dimensional vector representing values of all the features, and y_i^s is a single value as the label/output. Regarding any target link $a \in A'$, a test set of M samples D_a^t is obtained for link a from its CL data records. We use $x_{a,i}^t$ denotes features of sample i in D_a^t. Note that we do not have any label value for $x_{a,i}^t$.

Fig. 10 shows the flowchart for the zero-shot transfer learner.

5.2.1. Z-score data standardisation

Data standardisation is an important component for the convergence of the method (Anysz, 2016). Conventional methods, such as *z*-score standardisation and min-max standardisation, have been widely used in data mining. Considering that the min-max standardisation approach is susceptible to outliers, the *z*-score standardisation approach is adopted to scale the label of traffic flow in the link deduced from the LPR data. For all source links:

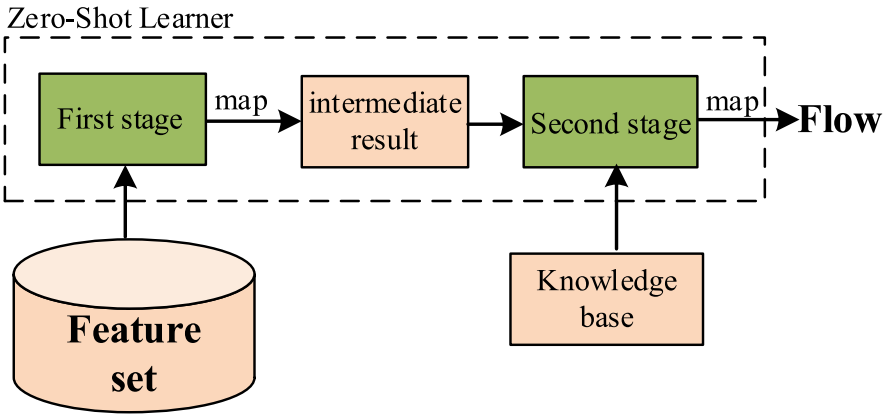


Fig. 10. Flowchart for the zero-shot transfer learner.

$$\bar{y}_v^s = \frac{y_v^s - \mu^s}{\sigma^s} \tag{7}$$

where y_v^s is the label of a sample v in D^s, and $\bar{y}_v^s$ is the standardised label. μ^s and σ^s are the mean and standard deviation of all the labels in D^s respectively.

Note that for a machine learning method, three sequential steps are usually needed: training, validation, and estimation. All samples are standardised together, and then divided into training set and validation set; *i.e.*, μ^s and σ^s are calculated based on all samples, and are the same for the training and validation sets.

5.2.2. Re-scaled problem

Due to the z-score standardization, outputs of the model in the validation step need to be re-scaled by multiplying σ^s and then adding μ^s. Let $(x_v^s, y_v^s, \bar{y}_v^s)$ denote a sample v from the validation set. Then, v should be re-scaled as:

$$\bar{y}_v^s = \mathrm{F}(x_v^s) \tag{8}$$

$$y_v^s = \mathrm{L}^s(\bar{y}_v^s) = \bar{y}_v^s \sigma^s + \mu^s \tag{9}$$

where $\bar{y}_v^s$ is the intermediate output of the first stage. The trained model in from the source link can be used for flow estimation at target link $a \in A'$. Target link $a \in A'$ has M CL data records, which are closed in set D_a^t, and for each record $i \in D_a^t$:

$$\bar{y}_i^t = \mathrm{F}(x_i^t) \tag{10}$$

Traffic flow y_i^t can thus be obtained by:

$$y_i^t = L^t(\bar{y}_i^t) = \bar{y}_i^t \sigma_a^t + \mu_a^t \tag{11}$$

where parameters σ_a^t and μ_a^t are standard deviation and mean values of the link flows in target link $a \in A'$, which are however unknown because we have no LPR data for a. Therefore, it is essential to approximate the value of σ_a^t and μ_a^t. Two cases are addressed here.

Case 1.. *Empirically, we know that, for the same link at the same time of day, the mean values of flows are usually similar, thus the value of μ_a^t can be approximated by real surveys in the same context (same link at the same time in a day) but on different days.*

As per Case 1, suppose that we can get the value of μ_a^t based on real surveys. For the estimation of σ_a^t, $a \in A'$, only those samples in D^s with similar properties to $a \in A'$ should be used. For example, when estimating the traffic flow on a busy arterial road during peak hours, source links with light traffic are not helpful. We use $\bar{D}^a$ to denote the set of similar links to link $a \in A'$, clearly $\bar{D}^a \subseteq D^s$, and $\bar{\mu}_a$ is defined as the mean value of all these samples. Based on $\bar{\mu}_a$, σ_a^t can be approximated by $\sigma^s \cdot \frac{\mu_a^t}{\bar{\mu}_a}$. Therefore:

$$\begin{aligned} y_a^t &\approx F(x_a^t) \cdot \sigma^s \cdot \left(\frac{\mu_a^t}{\bar{\mu}_a}\right) + \mu_a^t \\ &= \frac{(F(x_a^t) \cdot \sigma^s + \mu^s)\mu_a^t}{\bar{\mu}_a} \\ &= L^s(F(x_a^t)) \cdot \frac{\mu_a^t}{\bar{\mu}_a} \\ &= H^s(x_a^t) \cdot \frac{\mu_a^t}{\bar{\mu}_a} \end{aligned} \tag{12}$$

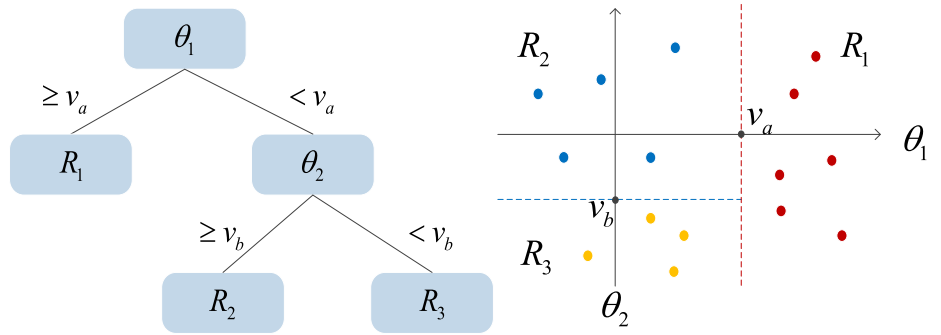


Fig. 11. Decision tree.

$H^s(\cdot)$ denotes the learner trained by source links, see (4), therefore, we can use the learner trained by a source link to estimate the target link flow.

Regarding the estimation of $\bar{\mu}_a$, the focus is to find a reasonable set $\bar{D}^a \subseteq D^s$. As an innovative part of the tailored machine learning model, we propose the following method to obtain $\bar{D}^a$:

The development of decision trees can shed some light on this problem. For the random forest model used in Section 4, each resultant decision tree can provide evidence for the categorization of the raw data. For example, the simple decision tree shown in Fig. 11(a) has three leaf nodes, R1, R2, and R3. All input data will be categorised into any of these three leaf nodes. Samples in the same leaf node have similar properties. Based on this, we build $\bar{D}^a$ as described below.

With a decision tree in the random forest model, for each CL data record $i \in D_a^t$ at link $a \in A'$, we can find the leaf node to which it belongs. The other data in D^s that also belong to the same lead node can be identified. After searching all decision trees, we can group these data for $i \in D_a^t$, which is grouped in set $\bar{D}_i^a$. Therefore, $\bar{D}^a = \cup_{i \in D_a^t} \bar{D}_i^a$ and $\bar{\mu}_a$ is then defined as the mean value of samples in $\bar{D}^a$.

5.3. A "warping distance" measurement method for sequence matching

Case 2.. *Differing from Case 1, if real survey data are unavailable, the flow in other similar links can be used to approximate the value of* μ_a^t.

As per Case 2, we propose a new method to approximate μ_a^t by sequence similarity matching, namely matching a similar link from existing links and using its traffic capacity to approximate μ_a^t. The measurement method for matching a similar link is elaborated below.

Traditional methods usually measure similarity by computing distance between sequences: however, simply comparing distances may be problematic (Fig. 12). For any source link $a \in A$, we can aggregate the set S_a (see Section 4.1) in terms of the total number of users in a time slice. Suppose that there are z time slices, this gives an original sequence B of length z, where $B = \{b_1, b_2, ...,b_z\}$. The same operation can be performed for another target link $a \in A'$, which gives a sequence C, where $C = \{c_1, c_2, ...,c_z\}$.

We can see from the illustrative example in Fig. 13 that the two sequences generally have very similar shapes, but they do not synchronise along the time line. Therefore, calculating their similarity directly may lead to an erroneous conclusion of significant discrepancy between them, since misplaced points will result in a large distance between each other. Therefore, we use a "warping

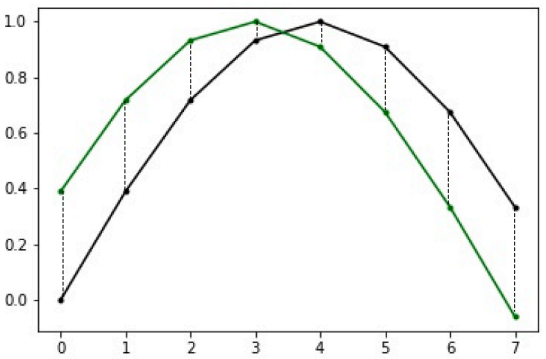


Fig. 12. Illustration of two similar sequences separated by a large distance.

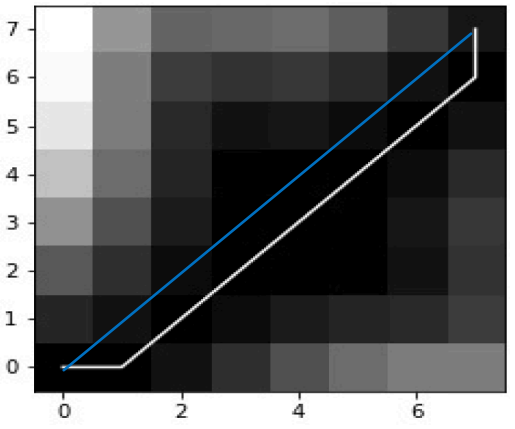


Fig. 13. Distance matrix.

distance" measurement method to assess the similarity between two sequences (Keogh and Pazzani, 2001). In this method, a $z \times z$ path matrix is created, and its element (i, j) denotes the distance between points b_i and c_j, such that $distance(b_i, c_j) = ||b_i - c_j||_2$, where $||\cdot||_2$ represents the l_2 norm.

The traditional similarity calculation method sums the anti-diagonal values in this matrix (*i.e.*, the blue line in Fig. 13). In the proposed method, we can find a warping path by minimising the cumulative distance between two sequences (*i.e.*, the white line in the Fig. 13). Several constraints are added here: (a) the starting/ending node of the warping path should be the first/last point of the path matrix; (b) the path advances one step at a time, while not decreasing. These constraints guarantee that if the path has passed element (i, j), then the preceding element can only be in one of the following three conditions: $(i-1, j-1)$, $(i-1, j)$, or $(i, j-1)$. The best match between two sequences is the one with the shortest path after aligning one sequence to the other. Thus, the warping distance $\eta(i, j)$ can be found using the recursive function given by:

$$\eta(i, j) = distance(b_i - c_j) + \min\{\eta(i-1, j-1), \eta(i-1, j), \eta(i, j-1)\} \tag{13}$$

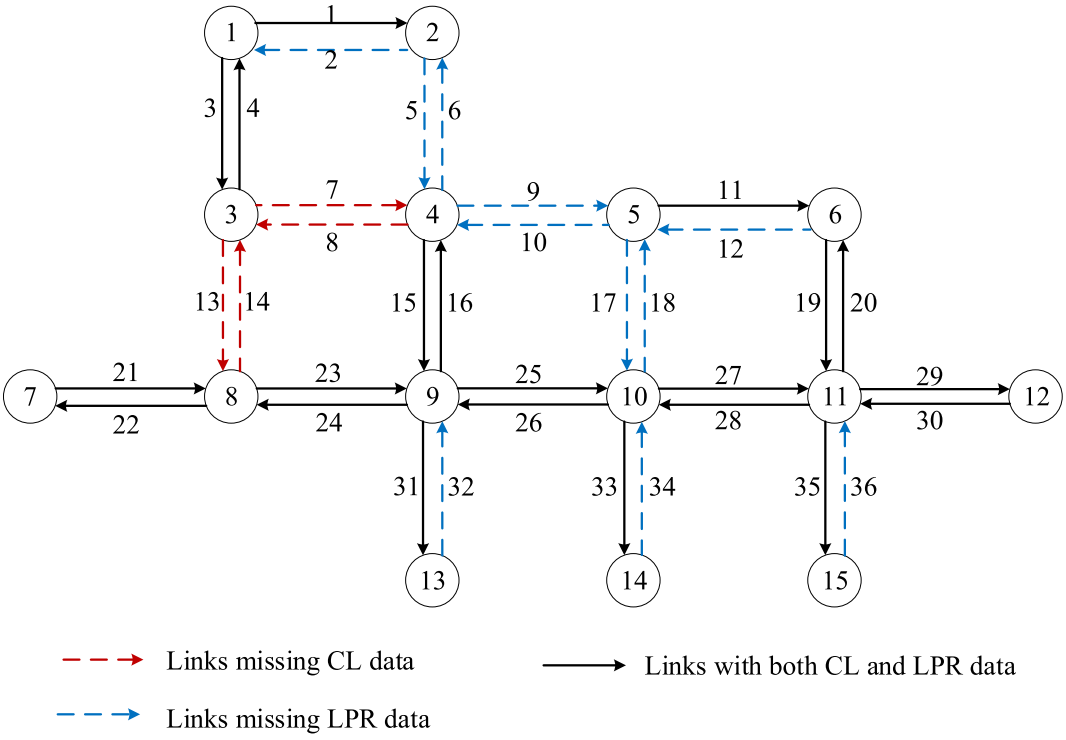


Fig. 14. Topologised network of the case study.

Table 4
List of models used in various modelling techniques.

Model/symbol	Description
RF	Random forest model
GBDT	Gradient boosting decision tree model
ST	Using spatio-temporal features
SW	Using sliding window method
MG	Using multi-grained method

where $\eta(i, j)$ denotes the warping path traversing node/element (i, j) in the matrix. The implication of this path is that it describes the mapping relationship of feature points from two time series. Take $\eta(2, 1)$ for instance, it means that b_2 should pair with c_1. We calculate the warping distance between the target link and all source links separately, and then select the average traffic flow of the most similar source link as the approximation for μ_a^t.

6. Case studies

The CL and LPR data collected for the study area shown in Fig. 1 are used in this section as case studies to verify the proposed models and methods. For the sake of presentation, this case study is topologised as that in Fig. 14, which contains 36 links and 15 nodes. The solid links are the 21 links with both CL and LPR data, *i.e.*, links in A. The period of this study is 8:00 am to 8:00 pm on weekdays. In the following two subsections, the two issues addressed at the end of Section 2 (single link and whole network) are discussed, respectively. Note that in the whole study area, there are also some links with no CL data (the red links), due to a problem of data collection or accessibility in the database of the telecommunications company, which are omitted in the case studies.

We use three different time intervals: 10-min, 30-min, and 60-min, to validate the performance of the models listed in Table 4. The distance threshold d_1 is set as 100 m, and the speed is divided into six categories between these vertex values: 0, 10, 20, 30, 40, 50, and above. In the extraction of multi-grained features, we use different size of sliding windows, which include: 1 min, 3 min, 5 min, 10 min, 20 min, 30 min, and 60 min (the size of a sliding window should be a divisor of the time interval and no longer than it).

The mean absolute percentage error (MAPE) is used to measure the performance of the models:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i} \tag{14}$$

where y_i is the observed traffic volume, and $\hat{y}_i$ is the estimated traffic volume for the i-th interval. N is the total number of estimated values.

It should be noted that, for model validation, only links with LPR data (black links in Fig. 14) have true values of traffic volume y_i, thus only the MAPE for these links can be calculated. Thus, without loss of generality, the concept of cross-validation is used here; namely, any of the black links could be taken as a target link. Then, using the models developed from the other source links, we can then estimate the flows on such a dummy target link, and test the performance of the proposed machine learning approach. Since the traffic volumes on these black links are known, it can also be used to test the two cases in Sections 5.2 and 5.3 (with given or known mean value μ_a^t).

6.1. Case study 1: single link flow estimation

As presented in Section 3, some modelling techniques are used here: Table 4 lists the symbols used in, and descriptions of, these modelling techniques. Different combinations of these techniques are used to build our machine learning models. We divide the data into 5-fold series for the cross-validation purposes.

To improve model performance, the parameter tuning criterion (Pedregosa et al., 2012) is used here for the parameters involved, *e.g.*, learning rate. The parameter tuning process is empirical, and we tuned some key parameters by grid search within the specified range. For the 10-min volumes, the best result has an MAPE of 9.63%. We tune four main parameters, the learning rate is set to 0.01, the maximum number of iterations is set to 320, the maximum depth of the individual regression estimators is set to 5, and the maximum number of splits is set to 16. For 30-min volumes, the lowest error is 7.62%, indicating a reasonable accuracy of the proposed method (the four parameters are 0.01, 390, 8, and 12, respectively). For 60-min volumes, we get the best result with an MAPE of 7.51% (the four parameters are 0.01, 350, 7, and 8, respectively).

Table 5 lists the performance of these models for the single link flow estimation (**Issue 1** in Section 2.2). We can see that the estimation error when using the 10-min interval is greater than that when using 30-min and 60-min intervals. The main reason for this was inferred to have been the influence of traffic signals, and the shorter 10-min interval is more affected by traffic signal timings, compared with the other two intervals.

Table 5
Performance of different models (MAPE).

Model	10-min interval	30-min interval	60-min interval
RF + ST	0.1037	0.0848	0.0974
RF + ST + SW	0.1015	0.0835	0.0859
RF + ST + SW + MG*	0.1014	0.0793	0.0813
GBDT + ST	0.1005	0.0839	0.0853
GBDT + ST + SW	**0.0963**	0.0785	0.0796
GBDT + ST + SW + MG*	0.0969	**0.0762**	**0.0751**

* The base learner of our proposed multi-grained scanning ensemble learning model can be RF or GBDT.

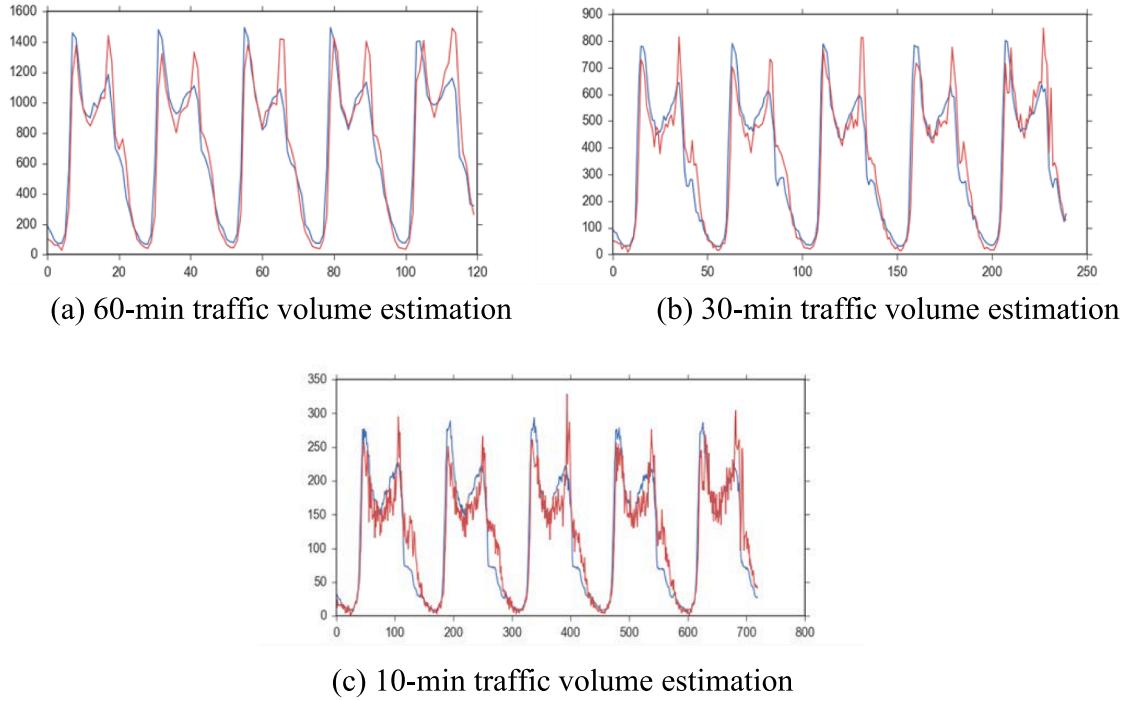


Fig. 15. Comparison between observation and estimation for 60-min volume, 30-min volume, and 10-min volume at link 25.

From the 30-min and 60-min cases, the use of sliding windows has increased the accuracy. Therefore, sufficient data are essential to the models: it is important to note that, even though the sliding windows can generate more samples and achieve better results, raw data remain essential. However, no significant effect was manifested in the 10-min case, which was mainly because the time interval was too short to extract useful features.

Since the information provided by the CL data is limited, the feature engineering we performed and the multi-grained method we applied have come close to the limiting accuracy, and hence better technology can only generate marginal improvements. Take ImageNet, the most famous competition in computer vision, as an example, the lowest error in top5 was 3% in 2016, after a year of rapid technological development, the error in the champion's solution dropped to 2.25% (Deng et al., 2009; Hu et al., 2018). In most machine learning tasks, although better methods have shown little improvement in accuracy, researchers are still actively studying them for huge cumulative benefits brought by such improvement.

6.2. Case study 2: transport network flow estimation

We then proceed to estimate the flow on those links with no LPR data (**Issue 2** in Section 2.2). Without loss of generality, link 25 is taken for cross-validation and demonstration purposes. In Fig. 15, the blue line shows the estimated traffic volume, and the red line shows the observed traffic volume (link 25).

Table 6
Performance of the model in terms of MAPE.

Link	10-min interval	30-min interval	60-min interval
1	0.180	0.174	0.261
3	0.253	0.131	0.056
4	0.271	0.213	0.030
11	0.466	0.096	0.175
15	0.390	0.246	0.083
16	0.208	0.205	0.429
19	0.358	0.247	0.029
20	0.412	0.262	0.062
21	0.334	0.091	0.283
22	0.181	0.267	0.176
23	0.250	0.108	0.089
24	0.153	0.097	0.207
25	0.320	0.142	0.162
26	0.176	0.196	0.023
27	0.615	0.180	0.052
28	0.155	0.255	0.018
29	0.541	0.094	0.090
30	0.209	0.095	0.063
31	0.370	0.133	0.310
33	0.211	0.108	0.079
35	0.144	0.083	0.054

Table 7
Matched links obtained by the warping distance method

Link	Matched link	Link	Matched link
1	30	24	19
3	28	25	11
4	28	26	11
11	31	27	16
15	25	28	16
16	28	29	19
19	31	30	15
20	4	31	11
21	4	33	22
22	33	35	11
23	19		

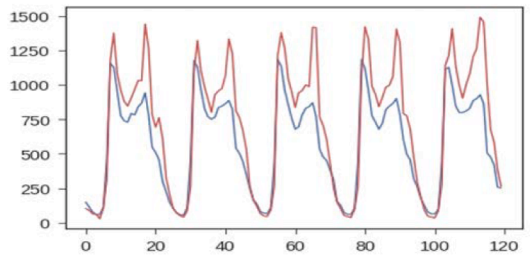
In Sections 5.2 and 5.3, we have proposed two different methods for the zero-shot learner of network flow estimation, for Cases 1 and 2, respectively. Thus, in this section, analysis of the performance of these two methods is described.

Firstly, for Case 1 in Section 5.2, we assume that a field survey was conducted at dummy target link 25, thus the mean value of its volume μ_a^t is obtained. Fig. 15 provides the estimated flow $\hat{y}_i$ (red lines) in contrast with true values y_i (blue lines) for each time interval, which demonstrated a sound match and evinced the good performance of the machine learning model. Similar to the single link case in Section 6.1, performance of the 60-min interval case is much better than that of the 30-min and 10-min intervals, attributable to the influences of traffic signals on CL data being more obvious when using 10-min intervals.

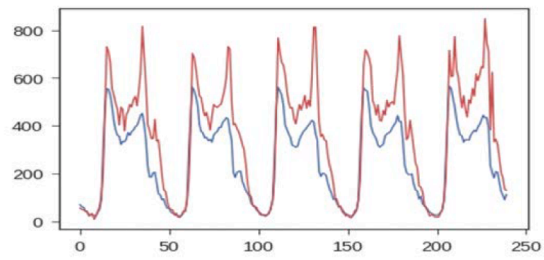
Table 6 lists the MAPE values for each tested link: the average MAPEs are 29.5%, 16.3%, and 13.0%, respectively. Considering that this is a zero-shot learning problem (to estimate the flow on links with no training samples), the model accuracy is quite good.

Secondly for Case 2 in Section 5.3, we assume that there is no survey data for the mean traffic volume μ_a^t, and a matching is needed between link 25 and any other source link. Therefore, the warping distance method proposed in Section 5.3 (see also Table 7) gives the matching results for each link in the study area. We can see that the matched link for link 25 is link 11. Thus, the flows in source link 11 are used to approximate the mean traffic volume on link 25.

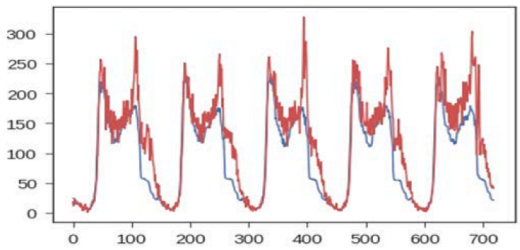
As shown in Fig. 16, performance of the second method in terms of Case 2 is however unsatisfactory, and is interior to that of the first method (Fig. 15). Such clear outperformance of the first method shows that the mean value μ_a^t is important to traffic flow estimation in an urban road network. Fig. 17 shows the estimated flow for the whole study area from 11:00 am to 12:00 am, which can be taken as the final output of the proposed tailored machine learning approach.



(a) 60-min traffic volume estimation



(b) 30-min traffic volume estimation



(c) 10-min traffic volume estimation

Fig. 16. Comparison between observed and estimated flows at link 25.

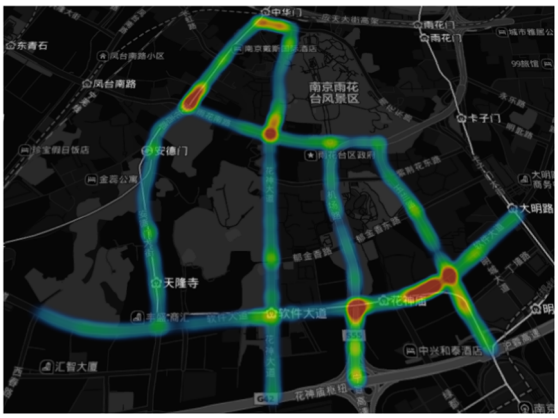


Fig. 17. Estimated flow for the study area.

7. Conclusion

Targeting at multi-source big data, we developed a tailored machine learning approach to approximate the traffic volumes by fusing CL and LPR data. The proposed model takes the spatio-temporal characteristic of the data into full consideration, which is convenient for practical implementation. Furthermore, to improve accuracy, we have added multi-grained features and used a sliding window to generate more samples. A zero-shot transfer learning model was proposed for the transport network flow estimation model, validated by a case study. For cases using 10-min intervals, the average MAPEs are 29.5%. For cases using 30-min intervals, the lowest error is 16.3%; while, when using 60-min intervals, the average MAPEs are 13.0%. Considering that this is a zero-shot learning task, namely to estimate the flow on links with no training samples, the model accuracy is quite good.

This research provides an initial step in the investigation of the use of multi-source data for estimating traffic volume. In future work, we will consider other location-based data and use them to estimate traffic volumes. More efforts are also needed to develop traffic volume estimation models using small data sets and big data sets, respectively. The semantic meaning-based fusion mechanism should also be investigated.

Acknowledgment

This study is supported by the National Key Research and Development Program of China (No. 2018YFB1600900).

Appendix A:. The sliding window method

For the CL data, we first aggregate the number of data records in each minute, thus for a specific day, there are 1440 slots for the whole 24 h. Hence, when a 10-min time interval is used, 144 samples will be generated for the whole 24 h each day; when the 60-min interval is taken, the number of samples each day is only 24. Thus, a sliding window method is used to extract more useful samples, as shown in Fig. 18. For example, when the 60-min interval is used, the window slides in a step length of 1 min, giving rise to 1381 samples (each contains 60 dimensions) per day to generate more samples.

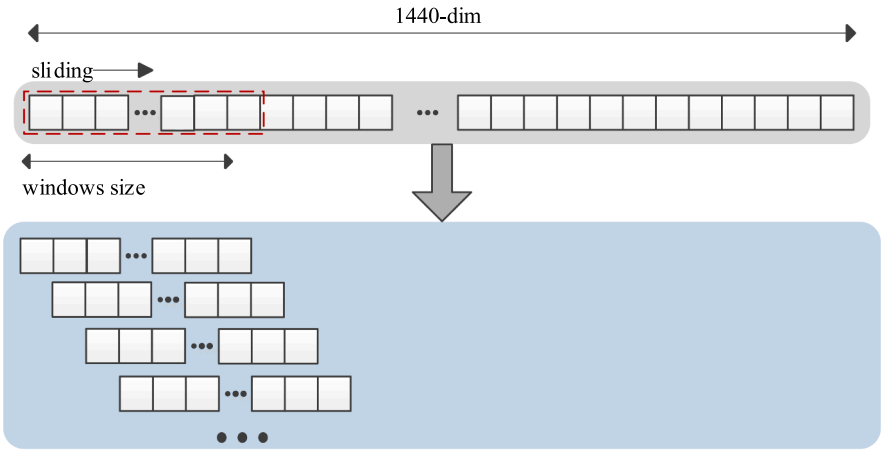


Fig. 18. Illustration of the sliding window method.

Appendix B:. List of features for the network-based flow estimation model

Taking the 30-min interval as an example, the full list of features is provided in the following table:

Features
Time slice ID
Hour of day
Day of week
Tail node of the link
Head node of the link
Tail node of the path
Head node of the path
Land use types
The number of lanes
The link direction
Number of users every minute (30 features)
Number of users whose average speed falls in 1st speed interval every minute
Number of users whose average speed falls in 2nd speed interval every minute
–
Number of users whose average speed falls in 6-th speed interval every minute
Number of users every 3 min (10 features)
Number of users whose average speed falls in 1st speed interval every 3 min
Number of users whose average speed falls in 2nd speed interval every 3 min
–
Number of users whose average speed falls in 6-th speed interval every 3 min
Number of users every 5 min (6 features)
Number of users whose average speed falls in 1st speed interval every 5 min
Number of users whose average speed falls in 2nd speed interval every 5 min
–
Number of users whose average speed falls in 6-th speed interval every 5 min
Number of users every 10 min (3 features)

Features
Number of users whose average speed falls in 1st speed interval every 10 min
Number of users whose average speed falls in 2nd speed interval every 10 min
–
Number of users whose average speed falls in 6th speed interval every 10 min
Number of users in the time slice
Number of users whose average speed falls in 1st speed interval in the time slice
Number of users whose average speed falls in 2nd speed interval in the time slice
–
Number of users whose average speed falls in 6th speed interval in the time slice

References

Alexander, L., Jiang, S., Murga, M., González, M.C., 2015. Origin–destination trips by purpose and time of day inferred from mobile phone data. Transport. Res. Part C: Emerg. Technol. 58, 240–250.

Alpaydin, E., 2014. Introduction to Machine Learning. MIT Press, Cambridge, MA.

Anysz, H., Zbiciak, A., Ibadov, N., 2016. The influence of input data standardization method on prediction accuracy of artificial neural networks. Procedia Eng. 153, 66–70.

Ban, X.J., Chen, C., Wang, F., Wang, J., Zhang, Y., 2018. Promises of Data from Emerging Technologies for Transportation Applications: Puget Sound Region Case Study (No. FHWA-HEP-19-026).

Ban, X.J., Hao, P., Sun, Z., 2011. Real time queue length estimation for signalized intersections using travel times from mobile sensors. Transport. Res. Part C: Emerg. Technol. 19 (6), 1133–1156.

Cascetta, E., 2009. Transportation systems analysis: models and applications. Springer Science & Business Media, New York, NY.

Chang, H., Lee, Y., Yoon, B., Baek, S., 2012. Dynamic near-term traffic flow prediction: systemoriented approach based on past experiences. IET Intel. Transport Syst. 6 (3), 292–305.

Chen, C., Ma, J., Susilo, Y., Liu, Y., Wang, M., 2016. The promises of big data and small data for travel behavior (aka human mobility) analysis. Transport. Res. Part C: Emerg. Technol. 68, 285–299.

Chen, D., 2017. Research on traffic flow prediction in the big data environment based on the improved RBF neural network. IEEE Trans. Ind. Inf. 13 (4), 2000–2008.

Chen, X.M., Zahiri, M., Zhang, S., 2017. Understanding ridesplitting behavior of on-demand ride services: an ensemble learning approach. Transport. Res. Part C: Emerg. Technol. 76, 51–70.

Daganzo, C.F., 1997. Fundamentals of Transportation and Traffic Operations, vol. 30 Pergamon, Oxford.

de Dios Ortuzar, J., Willumsen, L.G., 2011. Modelling Transport, 4th ed. Wiley, Hoboken, N.J.

Djukic, T., Van Lint, J.W.C., Hoogendoorn, S.P., 2012. Application of principal component analysis to predict dynamic origin-destination matrices. Transport. Res. Rec. 2283 (1), 81–89.

Dong, H., Wu, M., Ding, X., Chu, L., Jia, L., Qin, Y., Zhou, X., 2015. Traffic zone division based on big data from mobile phone base stations. Transport. Res. Part C: Emerg. Technol. 58, 278–291.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.F., 2009. ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision & Pattern Recognition.

Duan, L., Xu, D., Tsang, I., 2012. Learning with Augmented Features for Heterogeneous Domain Adaptation. ArXiv:1206.4660 [Cs]. Retrieved from http://arxiv.org/abs/1206.4660.

Gao, H., Liu, F., 2013. Estimating freeway traffic measures from mobile phone location data. Eur. J. Oper. Res. 229 (1), 252–260.

Han, J., Kember, J., 2000. Data Mining Concepts and Techniques. Morgan Kaufmann, Burlington, MA.

Herrera, J.C., Work, D.B., Herring, R., Ban, X.J., Jacobson, Q., Bayen, A.M., 2010. Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment. Transport. Res. Part C: Emerg. Technol. 18 (4), 568–583.

Ho, T., 2013. Urban Location Estimation for Mobile Cellular Networks: a Fuzzy-Tuned Hybrid Systems Approach. IEEE Trans. Wireless Commun. 12 (5), 2389–2399.

Huang, W., Song, G., Hong, H., Xie, K., 2014. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans. Intell. Transp. Syst. 15, 2191–2201.

Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks. Comput. Vision Pattern Recogn. 7132–7141.

Hunt, J.D., Simmonds, D.C., 1993. Theory and application of an integrated land-use and transport modelling framework. Environ. Plann. B 20 (2), 221–244.

Iqbal, M.S., Choudhury, C.F., Wang, P., González, M.C., 2014. Development of origin–destination matrices using mobile phone call data. Transport. Res. Part C: Emerg. Technol. 40, 63–74.

Jeong, Y.S., Byon, Y.J., Castro-Neto, M.M., Easa, S.M., 2013. Supervised weighting-online learning algorithm for short-term traffic flow prediction. IEEE Trans. Intell. Transp. Syst. 14 (4), 1700–1707.

Jin, X., Zhang, Y., Yao, D., 2007. Simultaneously prediction of network traffic flow based on PCA-SVR. In: International Symposium on Neural Networks, vol. 4. Springer-Verlag, pp. 1022–1031.

Jordan, M.I., Mitchell, T.M., 2015. Machine learning: Trends, perspectives, and prospects. Science 349 (6245), 255–260.

Keogh, E.J., Pazzani, M.J., 2001. Derivative dynamic time warping. In: Kumar, V., Grossman, R. (Eds.), Proceedings of the 2001 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, Philadelphia, PA, pp. 1–11. https://doi.org/10.1137/1.9781611972719.1.

Kotsiantis, S.B., 2007. Supervised machine learning: a review of classification techniques. Informatica 31 (3), 249–268.

Lam, W.H., Yin, Y., 2001. An activity-based time-dependent traffic assignment model. Transport. Res. Part B: Methodolog. 35 (6), 549–574.

Li, L., Zhang, J., Wang, Y., Ran, B., 2018. Missing value imputation for traffic-related time series data based on a multi-view learning method. IEEE Trans. Intell. Transp. Syst. 20 (8), 2933–2943.

Liu, Y., Jia, R., Xie, X., Liu, Z., 2018. A two-stage destination prediction framework of shared bicycles based on geographical position recommendation. IEEE Intell. Transp. Syst. Mag. 11 (1), 42–47.

Liu, Y., Liu, Z., Jia, R., 2019a. DeepPF: a deep learning based architecture for metro passenger flow prediction. Transport. Res. Part C: Emerg. Technol. 101, 18–34.

Liu, Y., Liu, Z., Vu, H.L., Lyu, C., 2019b. A spatio-temporal ensemble method for large-scale traffic state prediction. Comput.-Aided Civ. Infrastruct. Eng. https://doi.org/10.1111/mice.12459.

Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F., 2015. Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. 16 (2), 865–873.

Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y., 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transport. Res. Part C: Emerg. Technol. 54, 187–197.

Mehran, B., Kuwahara, M., 2013. Fusion of probe and fixed sensor data for short-term traffic prediction in urban signalized arterials. Int. J. Urban Sci. 17 (2), 163–183.

Ministry of Public Security, 2016. General technical specifications for intelligent monitoring and recording system for vehicles on roads. Ind. Stand. Publ. Safety, GA/T 497–2016.

Mo, B., Li, R., Zhan, X., 2017. Speed profile estimation using license plate recognition data. Transport. Res. Part C: Emerg. Technol. 82, 358–378.

Palatucci, M., Pomerleau, D., Hinton, G.E., Mitchell, T.M., 2009. Zero-shot learning with semantic output codes. In: Presented at the Advances in Neural Information Processing Systems, vol. 3872. Curran Associates Inc, Vancouver, B.C., Canada, pp. 1410–1418.
Panwai, S., Dia, H., 2005. Comparative evaluation of microscopic car-following behavior. IEEE Trans. Intell. Transp. Syst. 6 (3), 314–325.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Vanderplas, J., 2012. Scikit-learn: Machine learning in Python. J. Mach. Lear. Res. 12 (Oct), 2825–2830.
Polson, N.G., Sokolov, V.O., 2017. Deep learning for short-term traffic flow prediction. Transport. Res. Part C: Emerg. Technol. 79, 1–17.
Sekuła, P., Marković, N., Laan, Z.V., Sadabadi, K.F., 2018. Estimating historical hourly traffic volumes via machine learning and vehicle probe data: a Maryland case study. Transp. Res. Part C 97, 147–158.
Sheffi, Y., 1985. Urban Transportation Networks. Prentice-Hall Inc.
Sun, L., Lu, Y., Jin, J.G., Lee, D.-H., Axhausen, K., 2015. An integrated Bayesian approach for passenger flow assignment in metro networks. Transport. Res. Part C: Emerg. Technol. 52, 116–131.
Sun, S., Zhang, C., Yu, G., 2006. A bayesian network approach to traffic flow forecasting. IEEE Trans. Intell. Transp. Syst. 7 (1), 124–132.
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A., 2015. Going deeper with convolutions. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, pp. 1–9.
Wang, C., Mahadevan, S., 2011. Heterogeneous domain adaptation using manifold alignment. Presented at the Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, vol. 22, p. 1541.
Wang, F., Wang, J., Cao, J., Chen, C., Ban, X.J., 2019. Extracting trips from multi-sourced data for mobility pattern analysis: An app-based data example. Transport. Res. Part C: Emerg. Technol. 105, 183–202.
Wei, Y., Chen, M.C., 2012. Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. Transport. Res. Part C: Emerg. Technol.. 21, 148–162.
Wu, X., Guo, J., Xian, K., Zhou, X., 2018. Hierarchical travel demand estimation using multiple data sources: A forward and backward propagation algorithmic framework on a layered computational graph. Transp. Res. Part C 96, 321–346.
Wu, C., Thai, J., Yadlowsky, S., Pozdnoukhov, A., Bayen, A., 2015. Cellpath: Fusion of cellular and traffic sensor data for route flow estimation via convex optimization. Transport. Res. Part C: Emerg. Technol. 59, 111–128.
Yang, P., Zhu, T., Wan, X., Wang, X., 2014. Identifying significant places using multi-day call detail records. IEEE, Cyprus, pp. 360–366. https://doi.org/10.1109/ICTAI.2014.61.
Zhan, X., Li, R., Ukkusuri, S.V., 2015. Lane-based real-time queue length estimation using license plate recognition data. Transport. Res. Part C: Emerg. Technol. 57, 85–102.
Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X., 2016. DNN-based prediction model for spatio-temporal data, in the Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM Press, New York, NY, USA, pp. 1–4.