BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 12/15 pp171–187
DOI: 10.26599/BDMA.2023.9020017
Volume 7, Number 1, March 2024

PURP: A Scalable System for Predicting Short-Term Urban Traffic Flow Based on License Plate Recognition Data

Shan Zhang, Qinkai Jiang, Hao Li, Bin Cao, and Jing Fan*

Abstract: Accurate and efficient urban traffic flow prediction can help drivers identify road traffic conditions in real-time, consequently helping them avoid congestion and accidents to a certain extent. However, the existing methods for real-time urban traffic flow prediction focus on improving the model prediction accuracy or efficiency while ignoring the training efficiency, which results in a prediction system that lacks the scalability to integrate real-time traffic flow into the training procedure. To conduct accurate and real-time urban traffic flow prediction while considering the latest historical data and avoiding time-consuming online retraining, herein, we propose a scalable system for Predicting short-term URban traffic flow in real-time based on license Plate recognition data (PURP). First, to ensure prediction accuracy, PURP constructs the spatio-temporal contexts of traffic flow prediction from License Plate Recognition (LPR) data as effective characteristics. Subsequently, to utilize the recent data without retraining the model online, PURP uses the nonparametric method *k*-Nearest Neighbor (namely KNN) as the prediction framework because the KNN can efficiently identify the top-*k* most similar spatio-temporal contexts and make predictions based on these contexts without time-consuming model retraining online. The experimental results show that PURP retains strong prediction efficiency as the prediction period increases.

Key words: traffic flow prediction; *k*-Nearest Neighbor (KNN); License Plate Recognition (LPR) data; spatio-temporal context

1 Introduction

Traffic flow prediction is a procedure that utilizes traffic measurement data from monitoring devices installed near roads to predict the number of vehicles passing by in the future[1]. The information on short-term traffic flow prediction can be provided to drivers in real-time, providing them with a reliable estimation of travel state, expected delays, and alternative routes to their destinations[2, 3], which reduce the risk of accidents to a certain extent. Therefore, accurate real-time traffic flow prediction is the key to traffic induction and control, which directly affects the performance of the transportation system[4].

Real-time urban traffic flow prediction aims to efficiently predict traffic flow through urban intersections in a short period. Urban traffic flow prediction is more challenging than traffic flow prediction on a highway, because urban road networks have complicated environments and signal interference caused by dense buildings. In recent years, License Plate Recognition (LPR) data, which are collected from surveillance cameras installed at intersections, have been widely used for urban traffic flow prediction[5, 6]. LPR data can provide wider coverage and higher

• Shan Zhang, Qinkai Jiang, Hao Li, Bin Cao, and Jing Fan are with School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310000, China. E-mail: zhangshan@zjut.edu.cn; jiangqinkai@zjut.edu.cn; lihao@zjut.edu.cn; bincao@zjut.edu.cn; fanjing@zjut.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2022-11-27; revised: 2023-06-15; accepted: 2023-06-25

accuracy[7, 8] than other data sources, such as loop detector data[9, 10] and global positioning system data[11]. With a large amount of LPR data, mining the characteristics of traffic flows and efficiently conducting real-time urban traffic flow prediction have become considerable challenges.

The existing methods based on LPR data focus on the use of neural networks to ensure accurate or efficient prediction. For instance, the Long Short-Term Memory (LSTM) network is used to capture the temporal correlations of traffic states[12], and the Convolutional Neural Network (CNN) is used to extract spatial relationships[13]. However, for the real-time prediction of urban traffic flow, the existing methods are faced with a potential disadvantage, i.e., the lack of training scalability. Specifically, to derive accurate real-time prediction, integrating the recent traffic flow data into the prediction model is crucial. However, the existing methods need to retrain the prediction model from scratch when new traffic flow data are involved, and the training cannot be conducted in real-time, which leads to the dilemma between training time and real-time prediction. Hence, a prediction model that can be trained in a scalable manner is needed.

To this end, herein, we propose a scalable system for real-time Prediction of URban traffic flow based on license Plate recognition data (PURP). PURP aims to mine the effective characteristics of traffic flow from LPR data for accurate real-time traffic flow prediction while considering the latest historical data and avoiding time-consuming retraining online. First, PURP constructs the spatio-temporal contexts of traffic flow prediction as the effective characteristics. Specifically, given a location to be predicted, the temporal aspect of the spatio-temporal contexts captures the traffic flow correlation between the prediction duration and the historical observations, whereas the spatial aspect of the spatio-temporal contexts profiles the influence of nearby locations. Subsequently, in order to analyze the latest historical data without retraining the model, PURP uses the idea of the nonparametric method k-Nearest Neighbor (namely KNN)[14] as the prediction framework because the KNN focuses on predicting traffic flow by rapidly finding and utilizing similar historical data. Hence, PURP uses KNN to efficiently identify the spatio-temporal contexts of traffic flow prediction and make predictions based on these contexts without the time-consuming model training online. Moreover, to rapidly construct spatio-temporal contexts to ensure the prediction efficiency of PURP, we propose a circular array-based data structure to organize LPR data, which can provide $O(1)$ running time for data access.

Based on the aforementioned idea, initially, PURP transforms the traffic flow data by counting the license plates from LPR data and maintains the traffic flow data using the proposed data structure. Then, PURP constructs the spatio-temporal contexts of traffic flow prediction. Afterward, PURP builds traffic flow vectors[15] for the spatio-temporal contexts and identifies the top-k similar spatio-temporal contexts by calculating the similarity between the vectors. Finally, the traffic flow values corresponding to these contexts are used to make predictions.

In general, the contributions of this study can be summarized as follows:

(1) We propose PURP, a scalable system for predicting short-term urban traffic flow in real-time. PURP utilizes up-to-date historical data for real-time traffic flow prediction without the time-consuming model training online. Notably, when the prediction duration increases, the system response time of PURP remains stable, which ensures the prediction efficiency of the real-time prediction system.

(2) We propose a data structure based on queues and arrays to support the quick query for LPR data and efficient online construction of the spatio-temporal contexts for real-time traffic flow prediction.

(3) Extensive experimental evaluation reveals the prediction accuracy and efficiency of PURP. The experimental analysis also shows the scalability of PURP.

The remainder of this paper is organized as follows: Section 2 introduces the preliminaries and defines the problem. Section 3 presents an overview of the PURP system. Section 4 elaborates on the details of the first module in the PURP system. Section 5 describes the second module in the PURP system. Section 6 discusses the last module in the PURP system. Section 7 presents the experimental results. Section 8 reviews the related work. Finally, Section 9 concludes the paper.

2 Preliminary

In this section, we present a set of preliminaries that are

important to set the stage for understanding the PURP system. In particular, first, we introduce the concepts of checkpoint, traffic flow, target checkpoint, influential checkpoint, etc. Then, we formalize the problem to be addressed in this study. Finally, we introduce the underlying data structure. Table 1 summarizes the frequently used notations in this study.

2.1 Concept definitions

Checkpoint. Checkpoint c denotes the location where a surveillance camera is installed. The surveillance camera is used for real-time monitoring and recording throughout the day. The monitoring targets are the vehicles passing through this checkpoint. The main contents of monitoring and recording include various information regarding the vehicles, such as vehicle model, color, license plate number obtained by automatic identification, driving direction, and transit time. In urban roads, surveillance cameras are installed in every lane of intersections.

Traffic flow. We refer to the number of cars passing through a checkpoint within a specific period as traffic flow. In this study, the historical traffic flow is transformed by counting the license plates monitored by the surveillance cameras.

Target and influential checkpoints. Based on the definitions of checkpoint and traffic flow, we regard the checkpoint where the prediction is performed as target checkpoint c_{target}. The traffic flows of nearby checkpoints are expected to influence each other. The checkpoint within a certain area of the target checkpoint is regarded as the influential checkpoint c'_i, $i = 1, 2, \ldots, n$.

Influential and benchmark periods. We assume that the traffic flow in the past s minutes of a specific time point t is similar to that in the next period of the time point t. The past s minutes are referred to as the influential period. The request issuing time point is the time when the user issues the prediction query. When t is the request issuing time point, we refer to the influential period as the benchmark period. We use the traffic flow in the benchmark period to predict the traffic flow in the prediction period.

Example. As shown in Fig. 1, a user issues a request to "predict the traffic flow of the checkpoint c_{target} from 8:20 to 8:30" at 8:00, and we set s, i.e., the length of the influential period, to 30 min. Thus, the benchmark period is [7:30, 8:00].

2.2 Problem definition

Based on the understanding of the concepts of target checkpoint and traffic flow, PURP defines traffic flow prediction as follows:

Definition 1 Given a set of influential checkpoints C' and a set of historical periods H for target checkpoint c_{target}, we predict the future traffic flow for target checkpoint c_{target} from the start time t_{start} to the end time t_{end}.

2.3 Data structure

Traffic flow table *T*. PURP maintains a data structure, i.e., a traffic flow table, which is used to store the traffic flow of each checkpoint in the latest historical days d. We utilize an array to rapidly locate the traffic flow storage location of each checkpoint, because the

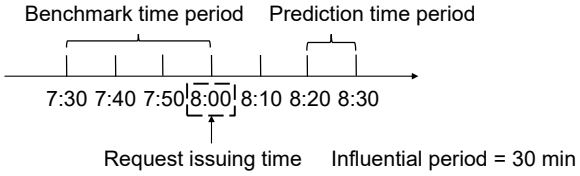


Fig. 1 Illustration of different periods.

Table 1 Frequently used notations.

Notation	Description
$C = \{c_i \mid i = 1, 2, \ldots, m\}$	Set of m checkpoints
c_{target}	Target checkpoint
r (km)	Radius of the circle centered on c_{target}
$C' = \{c'_i \mid i = 1, 2, \ldots, n\}$	Set of n influential checkpoints
s (min)	Influential period
b (min)	Benchmark period
v (min)	Time offset to move the benchmark period b to get the historical time period
d (day)	Historical days to store traffic flow values
$H = \{h_i \mid i = 1, 2, \ldots, (2v+1) \times d + v\}$	Set of historical time periods
T	Trafic flow table

time to access an element within an array is short. Each element stores a queue, which stores the traffic flow values every period in the latest d day. Herein, the length of the period is fixed at 1 min. We use a queue to store the historical traffic flow of each checkpoint because of its first in and first out characteristic, which means that we can delete records stored earlier and add new records according to time series to update the latest d day data. Notably, we use a circular array instead of a linked list to implement a queue to store historical traffic flow because the time complexity for accessing a linked list and a circular array is $O(n)$ and $O(1)$, respectively, which means that a circular array is faster than a linked list for querying traffic flow values from the table.

Example. Figure 2 shows an example of the traffic flow table. The left of Fig. 2 is an array where each entry corresponds to the historical traffic flow of a unique checkpoint. The right part of Fig. 2 is the stored content in each entry. The stored content in each entry is a queue to store the traffic flow every minute in the latest d day of each checkpoint.

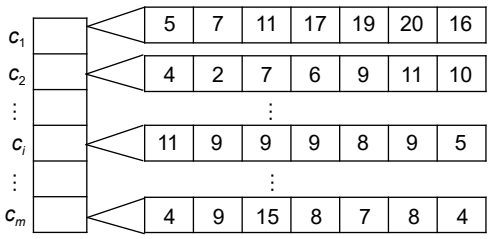


Fig. 2 Traffic flow table.

3 PURP Overview

In this section, we present an overview of PURP. PURP is a scalable system for the real-time prediction of short-term traffic flow at urban checkpoints. Based on the idea of KNN, PURP aims to identify the top-k spatio-temporal contexts of the target checkpoint to predict traffic flow. In order to achieve this goal, the PURP system comprises three main modules, namely, the traffic flow extraction module, the spatial and temporal context construction module, and the prediction module. The framework of the PURP system is shown in Fig. 3. The three modules are briefly described as follows:

In the traffic flow extraction module, PURP collects monitoring data from surveillance cameras at checkpoints, extracting traffic flow values from the data. The surveillance cameras record vehicle information at each timestamp; however, PURP aims to obtain the traffic flow values at checkpoints in continuous periods. In this module, PURP transforms the traffic flow data by counting the license plates from the recognition data and stores the traffic flow data in the traffic flow table. Because the traffic flow table stores the latest traffic flow data, with the generation of the monitoring data, PURP needs to update the traffic

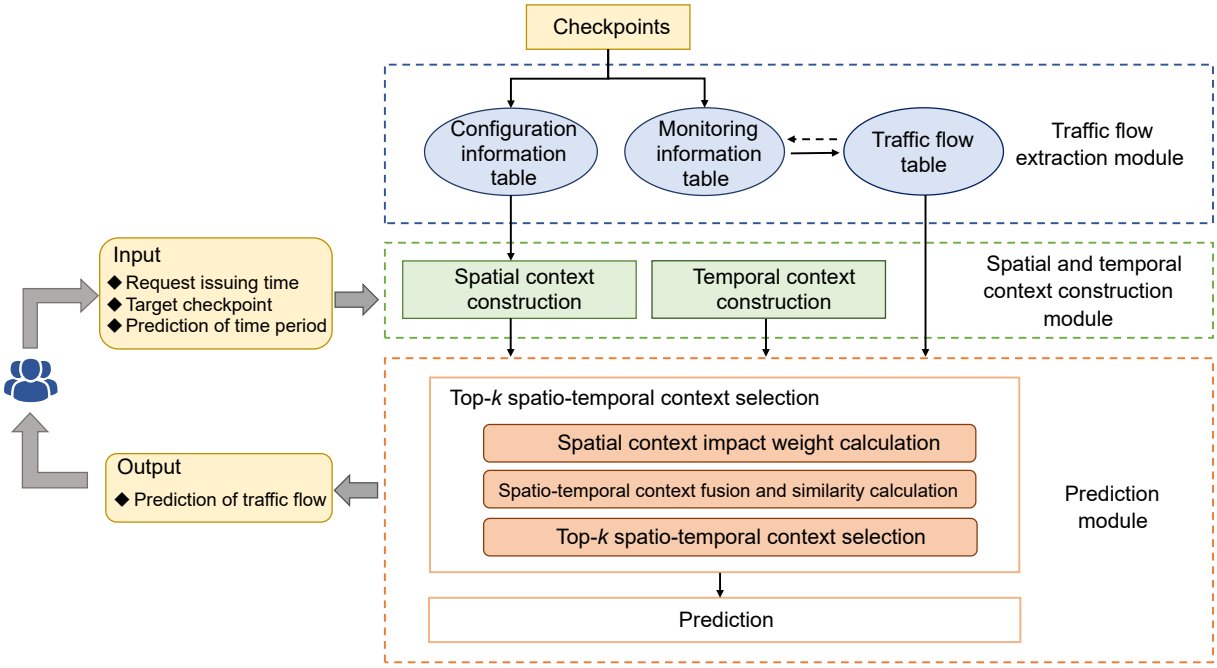


Fig. 3 PURP system.

flow table. As indicated by the dotted arrow in Fig. 3, PURP sends an update request to the monitoring information table. Then, PURP uploads the next period's traffic flow values for all checkpoints from the monitoring information table to the traffic flow table.

The goal of the spatial and temporal context construction module is to construct the spatial and temporal contexts of traffic flow prediction at the target checkpoint separately. Based on the checkpoint information obtained from the traffic flow extraction module, PURP uses the influential checkpoints of the target checkpoint to construct the spatial contexts. The historical periods of the benchmark period are used to construct the temporal contexts.

In the prediction module, PURP identifies the top-k spatio-temporal contexts and makes predictions based on these contexts. First, PURP calculates the impact weight of the spatial contexts of the target checkpoint. Then, PURP calculates the similarity of the spatio-temporal contexts obtained from the spatial and temporal contexts construction module. Afterward, PURP identifies the top-k most similar spatio-temporal contexts based on their similarity. Finally, PURP uses the top-k spatio-temporal contexts to predict traffic flow.

4 Traffic Flow Extraction

In the traffic flow extraction module, PURP collects monitoring data from the surveillance cameras and extracts traffic flow values from the data. The traffic flow table is used to store the traffic flow values. In this section, the specific operations of traffic flow extraction processing are as follows: First, we use the surveillance cameras to monitor information on the passing vehicles and obtain the following tables:

Configuration information table. This table has the following three attributes: (1) checkpoint ID, which is the unique camera identifier; (2) latitude, and (3) longitude, which describe the location information of the checkpoint.

Monitoring information table. Apart from the checkpoint ID, the table contains two other attributes: (1) the license plate number of the vehicle that passed by the checkpoint and was monitored by the surveillance camera and (2) the timestamp, i.e., the moment that the monitored vehicle passed by the checkpoint. Table 2 illustrates an example of the monitoring information table. The second row of the table implies that the vehicle with the license plate number XA00001 passed by checkpoint c_1 at 8:00:00.0 on September 1.

Table 2 Example of the monitoring information table.

Checkpoint ID	License plate number	Timestamp
c_1	XA00001	9-01 8:00:00.0
c_2	XA00002	9-01 8:00:01.0
c_2	XC00001	9-01 8:00:05.0
c_1	XB00004	9-01 8:01:09.2

PURP aims to assess the correlation between the traffic flows at the target and influential checkpoints in different periods. Thus, PURP counts the license plates to determine the traffic flow values every period of all checkpoints from the monitoring information table. Herein, the time granularity of a period is 1 min. To enable quick access to the traffic flow values in different periods, PURP stores the traffic flow values per minute in the latest d day of all checkpoints in the traffic flow table. The traffic flow table needs to store $60 \times 24 \times d \times n$ traffic flow values, where 60×24 are the traffic flow values in 1 day, and m is the number of all checkpoints.

The traffic flow table requires online updating in real-time to ensure that the traffic flow stored is up to date because we assume that the traffic situation in the latest historical period is nearly identical to that in the prediction period. PURP updates the traffic flow values every minute because the traffic flow table stores traffic flow values every minute. If the table is not full, PURP updates the table by adding new records to the end of the queues. If the table is full of d day's traffic flow values, PURP updates the table by deleting the records stored earliest and adding new records to the end of the queues. Figure 4 illustrates an example of the update process of the table.

Because PURP requires time to update the traffic flow table, it is possible that when a user sends a prediction request, the traffic flow table has not been

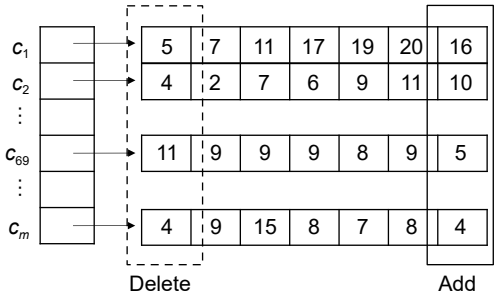


Fig. 4 Traffic flow table update.

updated. If PURP is updating the table when the prediction request is issued, PURP waits for the table to finish updating. Otherwise, PURP does not need to wait. We use the example shown in Fig. 5 to further illustrate the process. In the example, we assume that the time needed to update the traffic flow table is "3 s". PURP uploads the traffic flow value in [7:58-7:59] at 7:59:00 and completes the update at 7:59:03.

If the prediction request issuing time is between 7:59:00 and 7:59:03, PURP waits for the traffic flow table to finish updating. If the prediction request issuing time is between 7:59:03 and 8:00:00, PURP does not wait because the traffic flow table has been updated.

5 Spatial and Temporal Context Construction

PURP aims to construct the spatio-temporal contexts to analyze the correlation in both time and space for the traffic flow at the target checkpoint. In the spatial and temporal context construction module, PURP separately constructs the temporal and spatial contexts. When a user issues the prediction request, PURP obtains the prediction information, including the request issuing time, target checkpoint, and prediction period. Subsequently, PURP constructs the temporal and spatial contexts based on the prediction and configuration information from the traffic flow extraction module. For ease of explanation, we use an example that is set as follows: the request issuing time is "9-01, 8:00," the prediction period is [9-01, 8:05–8:06], and the influential period is 3 min.

(1) Temporal context construction

The traffic flow of a checkpoint is also correlated with its historical observation. PURP aims to construct the temporal contexts of the target checkpoint in the benchmark period online in this part. First, according to the request issuing time and influential period, PURP determines that the benchmark period is [7:57–8:00]. We assume that the traffic conditions within v minutes before and after the benchmark period are similar to the traffic condition in the benchmark period. Based on this assumption, PURP uses the benchmark period as a window and offsets the window forward and backward by v minutes (the moving unit is 1 min). Given that PURP offsets the window by one unit and obtains one period, we derive $2 \times v + 1$ periods as the historical periods of a historical day. For historical days d, we derive $(2 \times v + 1) \times d$ historical periods. Finally, we use the $((2 \times v + 1) \times d + v)$ historical periods that consider the day the request is issued as the temporal contexts.

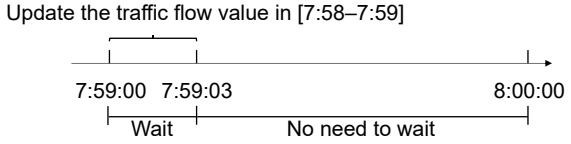


Fig. 5 Update time of traffic flow table.

Example. As depicted in Fig. 6, we set v to "1 min" and historical days d to "3 day". PURP derives $2 \times 1 + 1 = 3$ historical periods in the previous day, August 31, which are [8-31, 7:56–7:59], [8-31, 7:57–8:00], and [8-31, 7:58–8:01]. In the same manner, PURP derives the historical periods of August 29 and 30. However, for September 1, PURP can only shift the benchmark period back by 1 min and derive the historical period [9-01, 7:56–7:59]. Finally, PURP obtained $3 \times 3 + 1 = 10$ historical periods as the temporal contexts.

(2) Spatial context construction

In terms of space, we assume that the traffic conditions of the target and influential checkpoints are occasionally connected. This intuition originates from the observation of real scenarios, i.e., the congestion of one checkpoint will be transferred to a nearby checkpoint. Based on this idea, we analyze the region with radius r that assumes the target checkpoint as its center and define all checkpoints within this region as influential checkpoints. Notably, we use the R-tree[16] to efficiently identify the checkpoints within r km of the target checkpoint. We use the latitude and longitude information of the checkpoints to distribute all checkpoints in a two-dimensional coordinate system

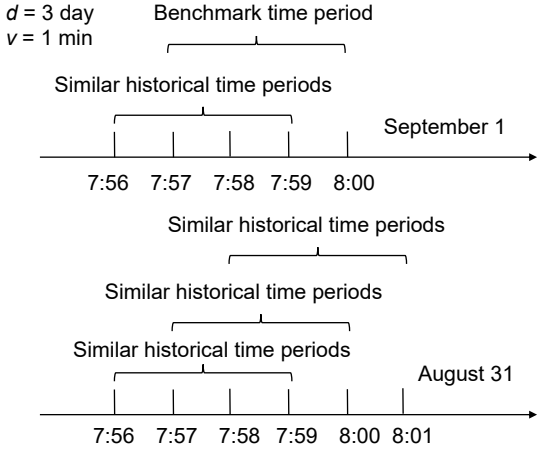


Fig. 6 Illustration of temporal context of [7:57–8:00].

and construct the R-tree with minimum bounding rectangles that contain two checkpoints with the smallest distance. Given that the radius parameter r is determined in advance, we can complete the influential checkpoints query of each checkpoint using the R-tree offline.

6 Predictions Based on Spatio-temporal Contexts

We aim to predict the traffic flow at the target checkpoint by considering the temporal and spatial effects. From the spatial and temporal context construction module, PURP obtains the temporal and spatial contexts, which are collectively referred to as spatio-temporal contexts. In the prediction module, in accordance with the idea of KNN, PURP aims to identify the top-k most similar spatio-temporal contexts at the target checkpoint. Subsequently, the traffic flow of these contexts is used to predict the traffic flow in the request issuing time. Therefore, traffic flow prediction is divided into two phases: (1) Top-k spatio-temporal context selection. In this phase, PURP calculates the similarity of the spatio-temporal contexts and identifies the top-k most similar spatio-temporal contexts based on the similarity. (2) Prediction. Based on the top-k spatio-temporal contexts, PURP predicts the traffic flow in the prediction request in this phase.

6.1 Top-k spatio-temporal context selection

In this phase, PURP aims to identify the top-k most similar spatio-temporal contexts. The four steps in identifying the top-k spatio-temporal contexts are as follows: traffic flow vector construction, influential checkpoint weight value calculation, spatio-temporal context similarity calculation, and top-k similar spatio-temporal context selection.

(1) Traffic flow vector construction

PURP calculates the similarity of the traffic flow vectors of the spatio-temporal contexts, identifying the top-k similar contexts. Herein, the traffic flow vector comprises the traffic flow value per minute in the historical period, and its dimension a depends on the length of the influential period s. For instance, we assume that the traffic flow values of c_1 in [9-01, 7:55–7:56], [9-01, 7:56–7:57], and [9-01, 7:57–7:58] are 6, 5, and 4, respectively. Thus, the vector of checkpoint c_1 in [9-01, 7:55–7:58] is $<6,5,4>$. In this step, PURP builds the traffic flow vectors for the traffic flow at the target and influential checkpoints in the historical periods.

To generate the traffic flow vectors, PURP needs to query the traffic flow value in each time segment (i.e., 1 min) of the historical period from the traffic flow table. Because the traffic flow table uses circular queues to store the traffic flow values per minute in chronological order, PURP aims to offset the tail node backward to locate the query time segment, where the offset distance is the difference between the latest time segment (stored in the tail node) and the query time segment. However, the two cases for the storage location of the query time segment and the latest time segment using a circular array are as follows: (1) the query time segment is before the latest time segment, and (2) the query time segment is after the latest time segment. These cases imply that the index of the query time segment cannot be obtained by offsetting the difference between the latest time segment and the query time segment.

Example. Figure 7a depicts an example that illustrates the storage location of the latest time segment and the query time segment. The queue records the traffic flow value per minute from 8:00 to 8:06. The tail node stores the traffic flow value in the latest time segment, i.e., [8:05, 8:06]. The query time segment is [8:01, 8:02]. In Fig. 7b, the query time segment is after the latest time segment. Therefore,

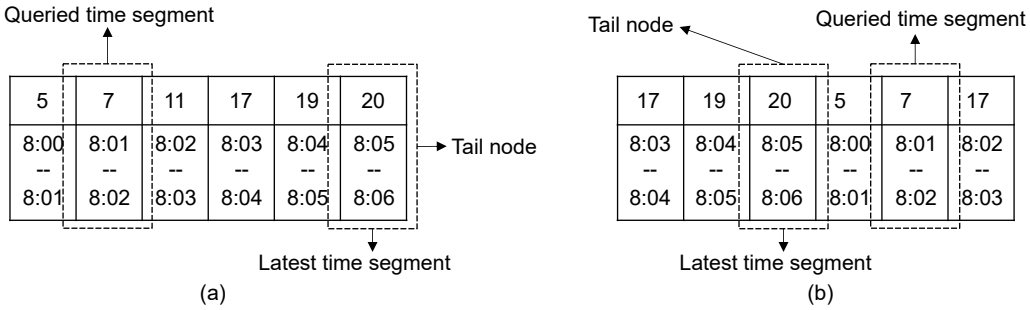


Fig. 7 Locations of the query time segment and the latest time segment.

PURP cannot use the tail node to shift the difference between the latest time segment and the query time segment backward to determine the position of the query time segment.

Based on the two situations, the index of the query time segment can be obtained using the following formula:

$$\text{index} = \begin{cases} \text{len} + \text{tail} - (l - q), & \text{if tail} - (l - q) < 0; \\ \text{tail} - (l - q), & \text{else} \end{cases} \quad (1)$$

where len is the queue length, tail is the index of the tail node, l is the latest time segment, and q is the query time segment.

Example. As shown in Fig. 7b, the queue length is 6. The difference between the latest time segment and the query time segment is 4, and the tail index is 2, which is smaller than the difference. Thus, the index of the query time slice is 4 × (6 + 2 − 4), which can be calculated using Eq. (1).

(2) Influential checkpoint weight value calculation

Different influential checkpoints exert a varying degree of impact on the target checkpoint. We determine the weight values between the influential checkpoint and the target checkpoint in the benchmark period by calculating the similarity of the vectors. Equation (2) is used to calculate the similarity between the vectors in the following:

$$\text{sim}\left(c_{\text{target}}(b), c_i'(b)\right) = \frac{\left|\sum_{d=1}^{\alpha} (\alpha^d)(\beta^d)\right|}{\sqrt{\sum_{d=1}^{m} (\alpha^d)^2}\sqrt{\sum_{d=1}^{m} (\beta^d)^2}},$$

where

$$\begin{aligned} \alpha^d &= V^d_{c_{\text{target}}(b)} - \bar{V}_{c_{\text{target}}(b)}, \\ \beta^d &= V^d_{c_i'(b)} - \bar{V}_{c_i'(b)} \end{aligned} \quad (2)$$

where $\text{sim}\left(c_{\text{target}}(b), c_i'(b)\right)$ is the degree of correlation between the target checkpoint c_{target} and its influential checkpoint c_i' in the benchmark period b, $V^d_{c_{\text{target}}(b)}$ is the value of d dimension in the traffic flow vector of $c_{\text{target}}(b)$, and $\bar{V}_{c_{\text{target}}(b)}$ is the average of each dimension's value in the traffic flow vector of $c_{\text{target}}(b)$. The value of $\text{sim}\left(c_{\text{target}}(b), c_i'(b)\right)$ ranges from 0 to 1. The higher the value of $\text{sim}\left(c_{\text{target}}(b), c_i'(b)\right)$, the greater the impact of c_i' on c_{target}. The influence weight of an influential checkpoint on the target checkpoint can be obtained by

$$w_{c_i'} = \frac{\text{sim}\left(c_{\text{target}}(b), c_i'(b)\right)}{\sum_{i=1}^{n} \text{sim}\left(c_{\text{target}}(b), c_i'(b)\right)} \quad (3)$$

where $\sum_{i=1}^{n} \text{sim}\left(c_{\text{target}}(b), c_i'(b)\right)$ is the sum of the similarity between all influential checkpoints and the target checkpoint, and $w_{c_i'}$ is the influence weight of the influential checkpoint c_i' that contributes to the target checkpoint.

Example. We assume that, for c_{target} in the benchmark period [9-01, 7:55–7:58], the traffic flow vector is (6, 5, 4). The influential checkpoint's (i.e., c_2' and c_3') traffic flow vectors are (6, 7, 5) and (5, 7, 6), respectively. In this manner, we can assess the similarity between each influential checkpoint and the target checkpoint, i.e., $\text{sim}\left(c_{\text{target}}, c_2'\right) = 0.5$ and $\text{sim}\left(c_{\text{target}}, c_3'\right) = 0.5$ (Eq. (2)); thus, the weights of c_2' and c_3' are 0.5 and 0.5 (Eq. (3)), respectively.

(3) Spatio-temporal context similarity calculation

In this step, PURP fuses the spatio-temporal contexts and calculates the similarity of the spatio-temporal contexts by considering (1) the similarity between the historical periods and the benchmark period of the target checkpoint and (2) the effect of different influential checkpoints,

$$\begin{aligned} F(h_j, b) &= \mu + \psi, \\ \mu &= \text{sim}\left(c_{\text{target}}(b), c_{\text{target}}(h_j)\right) \times P, \\ \psi &= \left(\sum_{i=1}^{n} \text{sim}\left(c_i'(b), c_i'(h_j)\right) \times w_{c_i'}\right) \times (1 - P) \end{aligned} \quad (4)$$

where $F(h_j, b)$ refers to the similarity between the spatio-temporal contexts which includes two parts μ and ψ. μ is the similarity of the target checkpoint c_{target} between a historical period h_j, and the benchmark period b, ψ is the similarity of influential checkpoints C' between h_j and b, which is used to consider the influence weight of C' to c_{target}. $\text{sim}\left(c_{\text{target}}(b), c_{\text{target}}(h_j)\right)$ indicates that PURP uses Eq. (2) to calculate the similarity between the traffic flow vector in the historical period and the benchmark period of c_{target}. $\sum_{i=1}^{n} \text{sim}\left(c_i'(b), c_i'(h_j)\right)$ is the sum of the similarity between the traffic flow vector in the historical period and the benchmark period of C'. $w_{c_i'}$ is the influence weight of the influential checkpoint to the target checkpoint in the benchmark period. P is an artificially regulated parameter that means the weight of the target checkpoint's traffic flow to $F(h_j, b)$. Based on the aforementioned operations, PURP obtains $(2 \times v + 1) \times d + v$ values because the number of h_j is $(2 \times v +$

$1)\times d+v$ F. For convenience, we denote $(2\times v+1)\times d+v$ as N.

(4) Top-k similar spatio-temporal context selection

We aim to sort the NF values obtained using the previous steps in descending order and select the top-k F. The spatio-temporal contexts corresponding to the top-k F values are the top-k similar spatio-temporal contexts. The method used to select the top-k F values is as follows: First, PURP initializes the min heap with size k. Subsequently, PURP calculates the F values and compares them with the root of the heap one by one. If the F value is larger than the root of the heap, PURP deletes the root and adds the F value to the heap. Finally, the values in the heap are the top-k F values.

The running time of the method is $O(N\times 2\times \log k)$, where the complexity of pushing or popping a value in the min heap is $\log k$, and PURP calculates NF values. Because the value of the parameter k is small (such as 20), the performance can be regarded as $O(N)$. The running time of the general sorting algorithm is at least $O(N\log N)$. Because F values are sorted online, efficiency needs to be improved by sorting rapidly.

Algorithm 1 shows the pseudocode for the top-k spatio-temporal context selection. We generate the traffic flow vectors for the target and influential checkpoints in the benchmark period (Lines 3–5). Initially, we calculate the similarities of the influential checkpoints to the target checkpoint (Line 6). Subsequently, we generate the traffic flow vectors for the target checkpoint in the historical periods and calculate the similarity between the historical periods and the benchmark period of the target checkpoint (Lines 8–9). Afterward, we generate the traffic flow vectors for the influential checkpoints in the historical periods and calculate the similarity between the historical periods and the benchmark period. Then, we calculate the weight values of the influential checkpoints and $F(h_j, b)$ (Lines 11–13). For each $F(h_j, b)$, we compare it with the root of the min heap. If $F(h_j, b)$ is larger than the root, we then delete the root and add $F(h_j, b)$ to the heap. Finally, we identify the top-k spatio-temporal contexts corresponding to the top-k F values that are in the min heap (Lines 14–17).

Algorithm 1 Top-k similar historical time periods selection

Input: Similar historical time periods $h_j, j = 1, 2, \ldots, (2v+1)\times d+v$; influential checkpoints $c_i, i = 1, 2, \ldots, n$; benchmark period b; target checkpoint c_{target}; traffic flow table T; value of parameter k and P

Output: Top-k spatio-temporal context

1 $V_{c_{\text{target}}(b)}$, $V_{c_i(b)}$, $V_{c_{\text{target}}(h_j)}$, $V_{c_i(h_j)}$ ⟵ null list;
2 heap ⟵ Initialize a min heap with an unordered array of k zeros by default;
3 $V_{c_{\text{target}}(b)}$ ⟵ Get traffic flow value per minute of c_{target} in b from T;
4 **for** each influential checkpoint c_i **do**
5 $V_{c_i(b)}$ ⟵ Get traffic flow value per minute of c_i in b from T;
6 sim1_i ⟵ sim ($c_{\text{target}}(b)$, $c_i(b)$); //Eq. (2)
7 **for** each historical time period h **do**
8 $V_{c_{\text{target}}(h_j)}$ ⟵ Get traffic flow value per minute of c_{target} in h_j from T;
9 sim2_j ⟵ sim ($c_{\text{target}}(b)$, $c_{\text{target}}(h_j)$); //Eq. (2)
10 **for** each influential checkpoint c_i **do**
11 $V_{c_i(h)}$ ⟵ Get traffic flow value per minute of c_i in h_j from T;
12 sim3_i ⟵ sim ($c_i(b)$, $c_i(h_j)$); //Eq. (2)
13 $w_{c_i} \longleftarrow \frac{\text{sim1}_i}{\sum_{i=1}^{n}\text{sim1}_i}$; //Eq. (3)
14 $F(h_j,b) \longleftarrow \text{sim2}_j\times P+(\sum_{i=1}^{n}\text{sim3}_i\times w_{c_i})\times(1-P)$; //Eq. (4)
15 **if** $F(h_j, b)$ > root of heap **then**
16 Delete the root;
17 Add $F(h_j, b)$ to heap;
18 **return** spatio-temporal context corresponding to values in heap

6.2 Prediction

Because the traffic flow in the benchmark period and the latest historical data are available, PURP obtains the top-k spatio-temporal contexts based on the benchmark period from the previous phases. However, PURP aims to make predictions based on the top-k spatio-temporal contexts in the prediction period. In this phase, PURP obtains the spatio-temporal contexts in the prediction period based on the contexts in the benchmark period and makes predictions.

First, PURP determines the historical periods of the k spatio-temporal contexts in the benchmark period. PURP offsets the end time of these historical periods forward by the offset distance to obtain k new start time. The offset distance is the prediction interval between the prediction period and the request issuing time. Based on the start time, PURP uses the duration of the prediction period as the period length and obtains k new periods. The spatio-temporal contexts corresponding to the new periods are the top-k spatio-temporal contexts in the prediction period. Afterward,

PURP obtains the traffic flow values in these k contexts by counting the traffic flow per minute from the traffic flow table. Finally, the average of the traffic flow values is taken as the traffic flow prediction value.

Algorithm 2 displays the pseudocode for prediction. Initially, we determine the prediction interval (between the request issuing time and the prediction period) and the duration of the prediction period. Subsequently, we offset the end time of the historical periods backward based on the prediction interval to determine the start time of the new periods. We use the duration as the length of the new periods and obtain k new periods (Lines 1−7). Afterward, we determine the traffic flow values in the k new periods by counting the traffic flow per minute from the traffic flow table (Lines 9−10). Finally, the average of the k traffic flow values is taken as the traffic flow prediction value (Lines 11−12).

Example. We suppose that the prediction period is [9-01, 8:05−8:06], the request issuing time is 9-01, 8:00, and the benchmark period is [9-01, 7:57−8:00]. We assume that PURP identifies the top-3 spatio-temporal contexts corresponding to the benchmark period. The three historical periods corresponding to the top-3 spatio-temporal contexts are [8-31, 7:56−7:59], [8-31, 7:57−8:00], and [8-31, 7:58−8:01]. First, PURP offsets the end times (i.e., "8-31, 7:59", "8-31, 8:00", and "8-31, 8:01") of the three historical periods forward by 5 min, which is the interval between the prediction period and the request issuing time. PURP uses three start times, i.e., "8-31, 8:04", "8-31, 8:05", and "8-31, 8:06". The duration of the prediction period is 1 min; thus, PURP uses "1 min" as the duration of the new period. Subsequently, PURP obtains three new periods, i.e., [8-31, 8:04−8:05], [8-31, 8:05−8:06], and [8-31, 8:06−8:07], which correspond to the spatio-temporal contexts of the target checkpoint in the prediction period. We assume that the corresponding traffic flow values of the three spatio-temporal contexts are 7, 8, and 9. The prediction value of traffic flow in [9-01, 8:05−8:06] for the target checkpoint is 8, which is the average of 7 + 8 + 9.

Algorithm 2 Prediction

Input: Top-k historical time periods $H = \{h_i | i = 1, 2, \ldots, k\}$ of target checkpoint c_{target}; traffic flow table T; prediction time period; request issuing time

Output: Prediction traffic flow

```
1  od ⟵ Starting time of prediction time period − the request issuing time;
2  length ⟵ Duration of the prediction time period;
3  count, sum ⟵ 0;
4  for each historical time period h_i ∈ H do
5      t_s ⟵ end time of h_i + od;
6      t_e ⟵ t_s + length;
7      New time period ⟵ [t_s, t_e];
8      for every minute e ∈ new time period do
9          value ⟵ get traffic flow value of c_target in e from T;
10         count ⟵ count + value;
11     sum ⟵ sum + count;
12 result ⟵ sum/k;
13 return result
```

7 Experiment

This section experimentally evaluates the performance of PURP on LPR data. First, we introduce the experimental settings. Second, we tune each parameter in PURP. Then, we compare the prediction efficiency and accuracy of PURP with those of the existing methods to demonstrate the effectiveness of PURP.

7.1 Experimental setup

Data preparation. All experiments in this section are based on a real data set. The data set is the real traffic flow data on the urban road network of Hangzhou City in China. In our experiments, the data were collected using 800 traffic surveillance cameras that were distributed at different checkpoints within the city road network. Each surveillance camera records the monitoring information of each passing vehicle. In total, 170 million records in terms of monitoring information over 34 consecutive day were obtained.

Two observations can be made from the data set: (1) The traffic flow for each checkpoint is regular every day. i.e., the period of maximum traffic flow is from 7:00 am to 9:00 am and from 4:30 pm to 6:30 pm because people go to work and get off work in the two periods. (2) The traffic flow on workdays and weekends is different, where the traffic peak during the workdays is more evident than that during the weekends. Therefore, our experiments separately consider the traffic flow in the peak periods on workdays and weekends. To avoid exceptional accident situations, we randomly select 10 checkpoints that have heavy traffic flow (i.e., the average traffic flow in 5 min is 200) during peak hours and predict the traffic flow at each checkpoint in the prediction periods. The prediction periods are selected from 7:00 am to 8:00 am and from 4:30 pm to 5:30 pm of 1 day in the historical data.

Based on the raw LPR data shown in Table 2, we count the traffic flow per minute at each checkpoint in the past 34 consecutive day. Furthermore, according to the longitude and latitude information of the checkpoint, we use the R-tree to obtain all adjacent checkpoints within a certain range of the checkpoint.

Comparison. We compare PURP with the AutoRegressive Integrated Moving Average (ARIMA)[17] and LSTM[18] models. ARIMA involves time series analysis for traffic flow prediction, and the LSTM network is used to capture the nonlinear dynamics in time series traffic data. We evaluate the performance of PURP and the other methods based on prediction accuracy and prediction response time. The response time refers to the time from when the user sends out the prediction demand to when the system returns the prediction result. All experiments are evaluated on a computer with Intel® Xeon® CPU E5-2637 3.50 GHz processor and 8 GB RAM with Windows 7. The programming language is Python 3.6, and the database is SQL Server 2014.

Accuracy measure. In the traffic system, the prediction accuracy of each checkpoint is equally important. Therefore, we employ the Mean Absolute Percentage Error (MAPE) as the accuracy measure. MAPE is a measurement of prediction accuracy for a prediction method in statistics that can eliminate the difference between checkpoints caused by the scale of traffic flow to evaluate the accuracy of each checkpoint equally. MAPE usually expresses the accuracy as a percentage and is defined by the following formula:

$$\text{MAPE} = \frac{1}{m \times n} \sum_{c=1}^{m} \sum_{t=1}^{n} \left| \frac{A_{\text{ct}} - F_{\text{ct}}}{A_{\text{ct}}} \right| \tag{5}$$

where A_{ct} is the actual value in prediction period t at checkpoint c, and F_{ct} is the forecast value. Herein, we use the MAPE value to represent the accuracy of the experimental results. For different experimental results, the smaller the MAPE value, the better the experimental result.

Hyperparameters. For the ARIMA model, the results of the optimal solution are that the parameter "autoregressive" is 5, the parameter "integrated" is 1, and the parameter "moving average" is 2. In the LSTM model, the network has a visible layer with one input, one output, and a hidden layer with four LSTM neurons. We set epochs to 1500 and batch size to 1, respectively.

7.2 Inside PURP

In this section, we adjust six parameters that appeared in the study in terms of prediction accuracy and prediction efficiency. The six parameters include influential period s, radius r, historical days d of the traffic flow stored in the traffic flow table, time offset v of the benchmark period, parameter P (which is the artificially regulated parameter in Eq. (4)), and parameter k of the top-k similar historical periods. When we tune one of the parameters, the other parameters are fixed. In every parameter tuning experiment, PURP predicts the traffic flow at the 10 checkpoints per minute from 7:00 am to 8:00 am and from 4:30 pm to 5:30 pm of the last day of the historical data and obtains $(60 + 60) \times 10$ values. In the accuracy evaluation, a total of 1200 prediction values of traffic flow are used to calculate the MAPE value. In the efficiency evaluation, the average of the 1200 system response times is considered the average response time. Finally, we choose the optimal parameters based on the results of the evaluation of both accuracy and efficiency.

Accuracy evaluation. In this set of experiments, we evaluate the effect of different parameters on the prediction accuracy of PURP. We divide the six parameters into two categories and conduct experimental analysis as follows: (1) Parameters for spatio-temporal context construction. The parameters influential period s, historical days d, time offset v, and radius r are used to construct the spatio-temporal contexts. As shown in Figs. 8a–8d, as the values of the four parameters increase, the MAPE values first decrease sharply and then tend to become stable. The reason is that if one of the four parameters increases, the spatio-temporal contexts increase. The more the spatio-temporal contexts are, the more spatial and temporal connections that we can find for traffic flow prediction at the target checkpoint, and the more accurate the prediction we can obtain. Moreover, as the spatio-temporal context increases further, it may contain meaningless context; thus, improving the prediction accuracy does not help. The results illustrate the effectiveness of considering the spatial and temporal contexts for traffic flow prediction. (2) Parameters for identifying the top-k spatio-temporal contexts. Parameters k and P are used to identify the top-k most similar spatio-temporal contexts. Figure 8e displays the results of parameter k. When parameter k

increases, the MAPE values initially decrease dramatically and then tend to be stable. This is similar to the results of the parameters for the spatio-temporal context construction. Parameter P is the weight of the temporal contexts of the target checkpoint for the spatio-temporal contexts. As shown in Fig. 8f, the optimal value for P is 0.8.

Efficiency evaluation. In this set of experiments, we count the system response time to evaluate the prediction efficiency effect of the six parameters. Based on the architecture of PURP, the system response time includes two parts, namely, the spatio-temporal context construction module and the prediction module. The update time of the traffic flow table constructed in the traffic flow information extraction module is included in the prediction module. In Fig. 9, the following observations are illustrated: (1) As shown in Figs. 9a, 9c, and 9d, we notice that the average response time increases for both modules as one of the three parameters (i.e., s, d, and v) increases. The reason is that the three parameters affect the construction of the temporal contexts in the traffic flow information extraction module. As one of the parameters increases, the temporal contexts increase, which further increases the cost of identifying similar spatio-temporal contexts in the prediction module. (2) The impact of tuning radius r is provided in Fig. 9b. The increment of radius r does not affect the spatio-temporal context construction module because PURP constructs the spatial contexts offline. The increment of radius r only increases the cost of the prediction module. (3) From Fig. 9e, we can conclude that when parameter k increases, only the response time of the prediction module increases because parameter k only affects the number of similar spatio-temporal contexts in the prediction module. As the proportional parameter in Formula (4), the parameter P does not affect the response time. In Fig. 9f, the fluctuation of the average response time caused by the increment of parameter P could be attributed to the inclusion of the update time of the traffic flow table in the prediction module.

Optimal parameters. Based on accuracy evaluation and efficiency evaluation, we choose the optimal solution for the six parameters. Because parameter P does not affect the response time, we choose the P value corresponding to the minimum MAPE value as the optimal solution. In Fig. 8e, when parameter P is 0.80, the MAPE value is the smallest, and the prediction accuracy is the best. The five other parameters can impact the prediction efficiency. To ensure the accuracy and efficiency of the prediction, we select the smallest value in which the MAPE value tends to be stable. For instance, when the influential period s is 11 min, the MAPE value tends to be stable at 18, and the average response time is the smallest (in terms of the value of s when the MAPE value tends to be stable at 18); thus, 11 is the optimal solution of

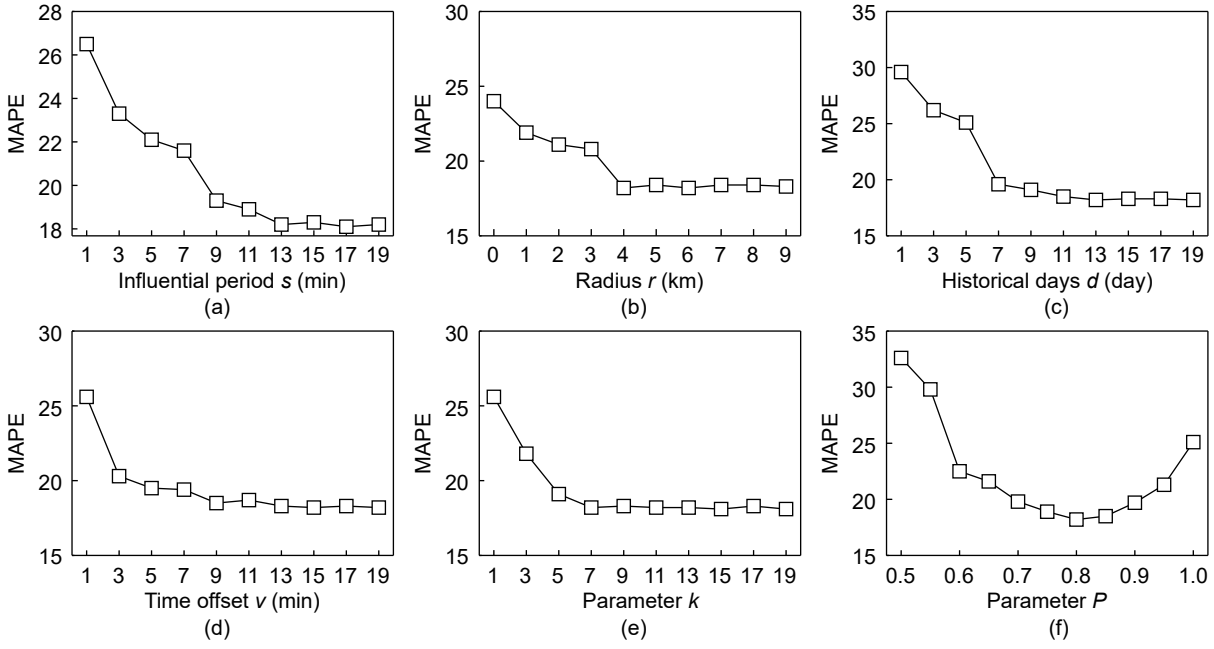


Fig. 8 Accuracy evaluation for prediction.

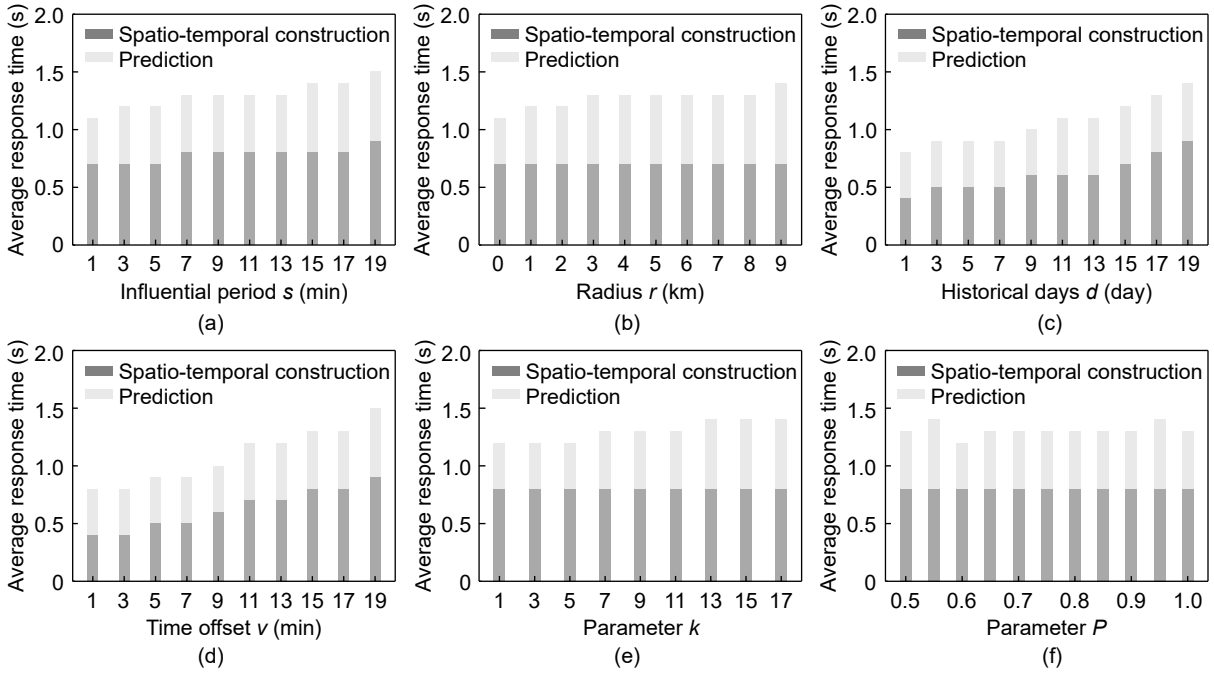


Fig. 9 Efficiency evaluation of the main modules of PURP.

influential period P. For PURP, based on the aforementioned evaluation, we use the following parameter settings: s = 11 min, r = 4 km, d = 11 day, v = 9 min, k = 7, and P = 0.8. Based on the optimal solution of the six parameters, PURP predicts 10 checkpoints per minute from 7:00 am to 8:00 am and from 4:30 pm to 5:30 pm, and records the update time of the traffic flow table. The average of 1200 values shows that the update time of the traffic flow table is 0.63 s.

7.3 Overall prediction accuracy and efficiency

This section investigates the overall prediction accuracy and efficiency of PURP by comparing PURP with the ARIMA and LSTM models. In terms of prediction efficiency and accuracy, we compare the three models by increasing the number of parameters, i.e., prediction interval between the prediction period and the request issuing time, duration of the prediction period, and peak period in different days, including workdays and weekends.

Two observations made in this study are as follows: (1) In terms of prediction accuracy, LSTM exhibits superior performance compared to the two other methods; however, the difference between PURP and LSTM is not obvious. Specifically, as depicted in Figs. 10a and 10b, as the prediction interval and duration increase, the results of PURP and LSTM tend to be stable, and the prediction effect of ARIMA becomes worse. The reason is that ARIMA iteratively

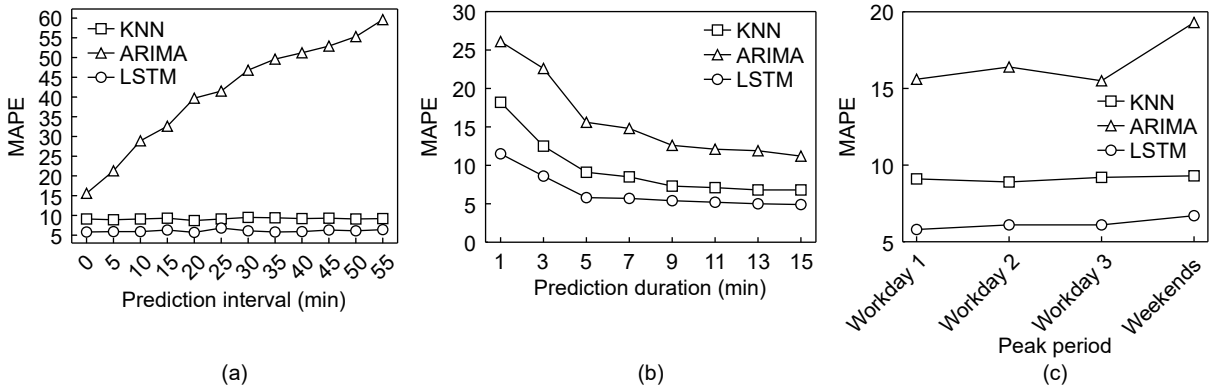


Fig. 10 Accuracy evaluation for prediction in different models.

predicts traffic flow, which suggests that the predicted value will be used as the training data for the prediction result of the next period. Inaccurate prediction results will lead to worse prediction results in the next stage. These results indicate that PURP can compete with LSTM. (2) In terms of prediction efficiency, PURP achieves optimal performance compared with the baselines. Figs. 11a and 11b depict the impact of increasing the prediction interval and duration. When the two parameters increase, the prediction efficiency of PURP remains stable and improves by at least 3 s compared with LSTM and ARIMA. The reason is that PURP can analyze the latest historical data without retraining the model online, whereas the two other methods need to retrain the models from scratch. The results indicate the powerful scalability of PURP. As shown in Fig. 11c, we predict the traffic flow in the peak periods on three workdays and one weekend. The results shown in Fig. 11 also illustrate that PURP has the best performance.

In this set of experiments, as a deep learning method, LSTM has exhibited strong prediction capability; meanwhile, the prediction capability of PURP is close to that of LSTM. The experiment results also reveal the powerful scalability of PURP, where the system maintains strong prediction efficiency regardless of the increase in prediction duration or prediction interval.

8 Related Work

Short-term traffic flow prediction has emerged as one of the major research fields in academia and industry, generating a large amount of work. Short-term traffic flow prediction can be classified according to the following aspects:

Time series-based prediction. Univariate time series forecasting of traffic flow is most common in the literature. Since the early 1980s, time series-based models, such as the Holt-Winters exponential smoothing model[19], have been widely used in traffic flow prediction. Several scholars focused on analyzing the impact of time series on traffic flow[20]. Liu et al.[21] investigated the effect of prediction interval on the prediction models for short-term traffic flow prediction. Zhang et al.[22] used a rescaled range method to estimate the long-range dependence of traffic time series and their conditional time series. The typical statistical methods used to predict traffic flow include the historical average approach[23], the ARIMA model[24, 25], and seasonal ARIMA[26]. In the past few years, deep learning methods have been widely used in prediction problems[27–29]. The LSTM-based architectures focus mainly on capturing the temporal dependencies of traffic states and exhibit powerful prediction capability[30–33]. However, as typical spatio-temporal data, the urban traffic flow at the prediction location is not only correlated with its historical observation but also influenced by its nearby locations. In this study, PURP constructs spatio-temporal contents to capture the spatio-temporal correlation of traffic states in the prediction location.

Spatio-temporal prediction. As typical spatio-temporal data[34, 35], traffic flow is characterized by randomness, time-varying, and spatial correlation. In the short-term traffic flow forecasting system, road occupancy, driving speed, and weather conditions will affect the traffic flow at the next moment[36]. The CNN and LSTM methods are used to capture the spatio-temporal features of traffic states[37–39]. To capture the spatio-temporal and temporal features of traffic and weather data, Sun et al.[40] proposed a hybrid CNN-LSTM model. However, deep learning methods that require a large amount of data for training models are unsuitable for real-time prediction systems. Training

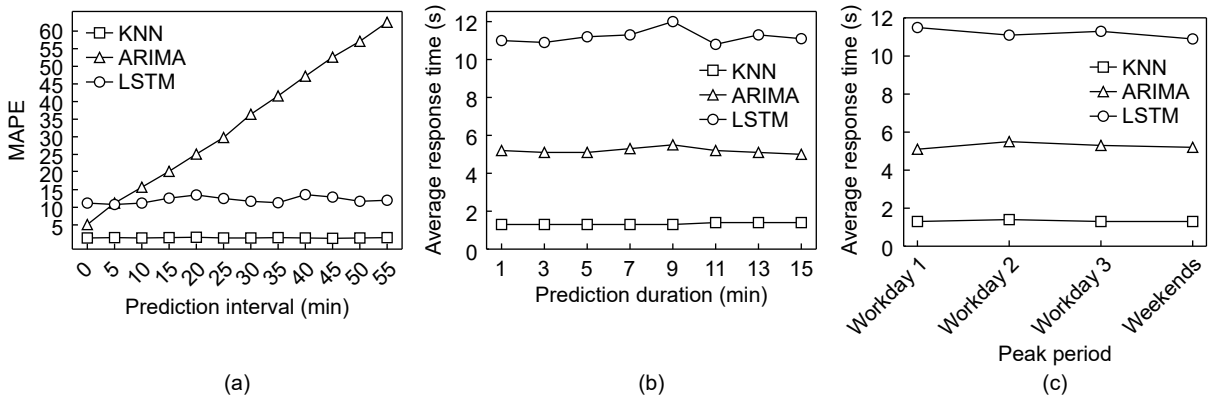


Fig. 11 Efficiency evaluation for prediction in three models.

models online is time-consuming and weakens system performance; meanwhile, using offline-trained models means that the latest historical data cannot be utilized in real-time, which will affect the accuracy of predictions. In this study, to analyze the latest historical data without the time-consuming retraining online, we use the idea of KNN as the prediction framework to design PURP for rapidly capturing and utilizing the spatio-temporal information of the target checkpoint to make predictions in real-time.

9 Conclusion

This study proposes PURP, a scalable short-term urban traffic flow prediction system. When a user sends a traffic flow prediction request to PURP, PURP predicts the traffic flow in the request based on LPR data. PURP uses the historical observation of traffic flow at the prediction location as the temporal context and the influential checkpoints as the spatial contexts to assess the correlation of the traffic flow prediction in terms of time and space. Then, PURP rapidly identifies the top-k spatio-temporal contexts to predict the traffic flow in the request online. We conduct extensive experiments based on real data. The behavior of PURP is adjusted using six tuning parameters based on accuracy and efficiency. The results demonstrate the strong performance of PURP for predicting short-term traffic flow. Although the LSTM model exhibits higher prediction accuracy than PURP, PURP has a higher prediction efficiency under the condition that the prediction accuracy is good, which is crucial for real-time prediction systems. Simultaneously, when the prediction demand (such as prediction duration) increases, the prediction efficiency of PURP remains.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 62072405 and 62276233) and the Key Research Project of Zhejiang Province (No. 2023C01048).

References

[1] A. Emami, M. Sarvi, and S. A. Bagloee, Short-term traffic flow prediction based on faded memory Kalman Filter fusing data from connected vehicles and Bluetooth sensors, *Simul. Model. Pract. Theory*, vol. 102, p. 102025, 2020.

[2] C. Guo, D. Li, G. Zhang, and M. Zhai, Real-time path planning in urban area via VANET-assisted traffic information sharing, *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 5635–5649, 2018.

[3] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, Adaptive multi-kernel SVM with spatial-temporal correlation for short-term traffic flow prediction, *IEEE Trans. Intell. Transport. Syst.*, vol. 20, no. 6, pp. 2001–2013, 2018.

[4] Z. Zheng, D. Wang, J. Pei, Y. Yuan, C. Fan, and F. Xiao, Urban traffic prediction through the second use of inexpensive big data from buildings, in *Proc. 25th ACM Int. Conf. Information and Knowledge Management*, Indianapolis, IN, USA, 2016, pp. 1363–1372.

[5] J. Tang, J. Zeng, Y. Wang, H. Yuan, F. Liu, and H. Huang, Traffic flow prediction on urban road network based on license plate recognition data: Combining attention-LSTM with genetic algorithm, *Transport-metrica A*: *Transport Science*, vol. 17, no. 4, pp. 1217–1243, 2021.

[6] J. Tang and J. Zeng, Spatiotemporal gated graph attention network for urban traffic flow prediction based on license plate recognition data, *Comput. Aided Civ. Inf.*, vol. 37, no. 1, pp. 3–23, 2022.

[7] J. Liu, F. Zheng, H. J. van Zuylen, and J. Li, A dynamic OD prediction approach for urban networks based on automatic number plate recognition data, *Transportation Research Procedia*, vol. 47, pp. 601–608, 2020.

[8] X. Zhan, R. Li, and S. V. Ukkusuri, Link-based traffic state estimation and prediction for arterial networks using license-plate recognition data, *Transportation Research Part C*: *Emerging Technologies*, vol. 117, p. 102660, 2020.

[9] K. Y. Chan and T. S. Dillon, On-road sensor configuration design for traffic flow prediction using fuzzy neural networks and Taguchi method, *IEEE Trans. Instrum. Meas.*, vol. 62, no. 1, pp. 50–59, 2013.

[10] J. Snowdon, O. Gkountouna, A. Züfle, and D. Pfoser, Spatiotemporal traffic volume estimation model based on GPS samples, in *Proc. Fifth Int. ACM SIGMOD Workshop on Managing and Mining Enriched Geo-Spatial Data*, Houston, TX, USA, 2018, pp. 1–6.

[11] B. Huang and Y. Bai, Short-term traffic flow prediction based on bus floating car, in *Proc. IEEE 9th Int. Conf. Electronics Information and Emergency Communication* (*ICEIEC*), Beijing, China, 2019, pp. 303–306.

[12] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, Long shortterm memory neural network for traffic speed prediction using remote microwave sensor data, *Transportation Research Part C*: *Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[13] C. Zheng, X. Fan, C. Wen, L. Chen, C. Wang, and J. Li, DeepSTD: Mining spatio-temporal disturbances of multiple context factors for citywide traffic flow prediction, *IEEE Trans. Intell. Transport. Syst.*, vol. 21, no. 9, pp. 3744–3755, 2020.

[14] T. Cover and P. Hart, Nearest neighbor pattern classification, *IEEE Trans. Inform. Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[15] S. Li, Z. Shen, and G. Xiong, A *K*-nearest neighbor locally weighted regression method for short-term traffic flow forecasting, in *Proc. 15th Int. IEEE Conf. Intelligent*

Transportation Systems, Anchorage, AK, USA, 2012, pp. 1596–1601.

[16] A. M. Hendawi, J. Bao, M. F. Mokbel, and M. Ali, Predictive tree: An efficient index for predictive queries on road networks, in *Proc. IEEE 31st Int. Conf. Data Engineering*, Seoul, Republic of Korea, 2015, pp. 1215–1226.

[17] H. Z. Moayedi and M. A. Masnadi-Shirazi, Arima model for network traffic prediction and anomaly detection, in *Proc. Int. Symp. Information Technology*, Kuala Lumpur, Malaysia, 2008, pp. 1–6.

[18] Z. Xie and Q. Liu, LSTM networks for vessel traffic flow prediction in inland waterway, in *Proc. IEEE Int. Conf. Big Data and Smart Computing* (*BigComp*), Shanghai, China, 2018, pp. 418–425.

[19] P. S. Kalekar, Time series forecasting using holt-winters exponential smoothing, *Kanwal Rekhi School of information Technology*, vol. 4329008, no. 13, pp. 1–13, 2004.

[20] A. Hasnat and F. I. Rahman, Short term traffic flow forecasting by time series analysis, in *Proc. Int. Conf. Planning, Architecture and Civil Engineering*, Rajshahi, Bangladesh, 2019, p. 126441.

[21] Z. Liu, X. Qin, W. Huang, X. Zhu, Y. Wei, J. Cao, and J. Guo, Effect of time intervals on K-nearest neighbors model for short-term traffic flow prediction, *PROMET*, vol. 31, no. 2, pp. 129–139, 2019.

[22] H. Zhang, H. Zhou, Z. Liu, and K. Dong, The long-term correlation of conditional time series of financial time series, in *Proc. Third Int. Symp. Intelligent Information Technology Application*, Nanchang, China, 2009, pp. 489–492.

[23] B. M. Williams, P. K. Durvasula, and D. E. Brown, Urban freeway traffic flow prediction: Application of seasonal autoregressive integrated moving average and exponential smoothing models, *Transport. Res. Rec.*, vol. 1644, no. 1, pp. 132–141, 1998.

[24] Y. Chen, L. Shu, and L. Wang, Poster abstract: Traffic flow prediction with big data: A deep learning based time series model, in *Proc. IEEE Conf. Computer Communications Workshops* (*INFOCOM WKSHPS*), Atlanta, GA, USA, 2017, pp. 1010–1011.

[25] G. Shi, J. Guo, W. Huang, and B. M. Williams, Modeling seasonal heteroscedasticity in vehicular traffic condition series using a seasonal adjustment approach, *J. Transp. Eng.*, vol. 140, no. 5, p. 04014012, 2014.

[26] N. F. Polson and V. O. Sokolov, Deep learning for short-term traffic flow prediction, *Transportation Research Part C*: *Emerging Technologies*, vol. 79, pp. 1–17, 2017.

[27] G. Dai, C. Ma, and X. Xu, Short-term traffic flow prediction method for urban road sections based on space-time analysis and GRU, *IEEE Access*, vol. 7, pp. 143025–143035, 2019.

[28] R. Liu, F. Hong, C. Lu, and W. Jiang, Short-term traffic flow prediction based on deep circulation neural network, *J. Phys.* : *Conf. Ser.*, vol. 1176, no. 3, p. 032020, 2019.

[29] Z. Qu, H. Li, Z. Li, and T. Zhong, Short-term traffic flow forecasting method with M-B-LSTM hybrid network, *IEEE Trans. Intell. Transport. Syst.*, vol. 23, no. 1, pp. 225–235, 2022.

[30] P. Poonia and V. K. Jain, Short-term traffic flow prediction: Using LSTM, in *Proc. Int. Conf. Emerging Trends in Communication, Control and Computing* (*ICONC3*), Lakshmangarh, India, 2020, pp. 1–4.

[31] C. Ma, G. Dai, and J. Zhou, Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM-BILSTM method, *IEEE Trans. Intell. Transport. Syst.*, vol. 23, no. 6, pp. 5615–5624, 2022.

[32] K. Chen, S. Zhao, and D. Zhang, Short-term traffic flow prediction based on data-driven Knearest neighbour nonparametric regression, *J. Phys.* : *Conf. Ser.*, vol. 1213, no. 5, p. 052070, 2019.

[33] H. Gao, B. Qiu, R. J. D. Barroso, W. Hussain, Y. Xu, and X. Wang, TSMAE: A novel anomaly detection approach for internet of things time series data using memory-augmented autoencoder, *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 2978–2990, 2023.

[34] L. Kuang, J. Zheng, K. Li, and H. Gao, Intelligent traffic signal control based on reinforcement learning with state reduction for smart cities, *ACM Trans. Internet Technol.*, vol. 21, no. 4, p. 102, 2021.

[35] X. Yang, Y. Zou, and L. Chen, Operation analysis of freeway mixed traffic flow based on catch-up coordination platoon, *Accid. Anal. Prev.*, vol. 175, p. 106780, 2022.

[36] A. A. Aburas, M. Mehtab, and Y. Mehtab, Cricket world cup predictions using KNN intelligent bigdata approach, in *Proc. 2018 Int. Conf. Computing and Big Data*, Charleston, SC, USA, 2018, pp. 18–22.

[37] A. Nigam and S. Srivastava, Macroscopic traffic stream variables prediction with weather impact using hybrid CNN-LSTM model, in *Adjunct Proc. 2021 Int. Conf. Distributed Computing and Networking*, Nara, Japan, 2021, pp. 1–6.

[38] K. Lin, X. Xu, and H. Gao, TSCRNN: A novel classification scheme of encrypted traffic based on flow spatio-temporal features for efficient management of IIoT, *Comput. Netw.*, vol. 190, p. 107974, 2021.

[39] F. Ding, Y. Zhang, R. Chen, Z. Liu, and H. Tan, A deep learning based traffic state estimation method for mixed traffic flow environment, *J. Adv. Transp.*, vol. 2022, p. 2166345, 2022.

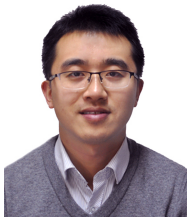
[40] B. Sun, W. Cheng, P. Goswami, and G. Bai, Flow-aware WPT k-nearest neighbours regression for short-term traffic prediction, in *Proc. IEEE Symp. Computers and Communications* (*ISCC*), Heraklion, Greece, 2017, pp. 48–53.

Shan Zhang received the BEng degree in computer science from Zhejiang University of Technology, China in 2017. She is now a PhD candidate at Zhejiang University of Technology. Her research interests are data mining and machine learning.

Hao Li received the BEng degree in computer science from Zhejiang University of Technology, China in 2020, where he is currently a master student. His research interests are data mining, time series forecast, and natural language processing.

Jing Fan received the BEng, MEng, and PhD degrees in computer science from Zhejiang University, China in 1990, 1993, and 2003, respectively. She is now a professor at School of Computer Science and Technology, Zhejiang University of Technology, China. She is a director of the China Computer Federation (CCF) and chairman of Chapter Hangzhou of CCF. Her current research interests include middleware, virtual reality, and visualization.

Qinkai Jiang received the BEng degree in computer science from Zhejiang University of Technology, China in 2018. He is now a master student at Zhejiang University of Technology. His research interests are data mining and machine learning.



Bin Cao received the PhD degree in computer science from Zhejiang University, China in 2013, he then worked as a research associate at Hongkong University of Science and Technology and Noah's Ark Lab, Huawei. He joined Zhejiang University of Technology, Hangzhou, China in 2014. His research interests include spatio-temporal databases and data mining.