6G-enabled short-term forecasting for large-scale traffic flow in massive IoT based on time-aware Locality-Sensitive Hashing

Fan Wang, Min Zhu, Maoli Wang*, Mohammad R. Khosravi, Qiang Ni, Shui Yu, and Lianyong Qi

***Abstract*—With the advent of the Internet of Things (IoT) and the increasing popularity of the Intelligent Transportation System, a large number of sensing devices are installed on the road for monitoring traffic dynamics in real-time. These sensors can collect streaming traffic data distributed across different traffic sites, which constitute the main source of big traffic data. Analyzing and mining such a big traffic data in massive IoT can help traffic administrations to make scientific and reasonable traffic scheduling decisions, so as to avoid prospective traffic congestions in the future. However, the above traffic decision-making often requires frequent and massive data transmissions between distributed sensors and centralized cloud computing centers, which calls for lightweight data integrations and accurate data analyses based on large-scale traffic data. In view of this challenge, a big data-driven and non-parametric model aided by 6G is proposed in this paper to extract similar traffic patterns over time for accurate and efficient short-term traffic flow prediction in massive IoT, which is mainly based on time-aware LSH (Locality-Sensitive Hashing). We design a wide range of experiments based on a real-world big traffic dataset to validate the feasibility of our proposal. Experimental reports demonstrate that the prediction accuracy and efficiency of our proposal are increased by 32.6% and 97.3%, respectively, compared with the other two competitive approaches.**

***Index Terms*—Short-term traffic forecasting, Intelligent Transportation System, time-aware LSH, massive Internet of Things, 6G, large-scale traffic management.**

I. Introduction

THE improvement of people's living standards has led to the expansion of data scale [1] and the growth of the number of vehicles. In response to this situation, the development of the Internet of Things (IoT) and mobile communication technologies render real-time traffic management feasible [2] [3]. First, a large number of devices (e.g., sensors) are installed on the road to monitor traffic dynamics in real-time. Afterwards, 6G technology enables frequent but stable traffic data transmission between these distributed sensors and the cloud platform. Finally, the large-scale traffic sensing data can be integrated to provide an effective reference for traffic management.

Nevertheless, congestions and queues occur more and more frequently nowadays, which requires traffic managers to develop more effective traffic management strategies based on the large-scale sensor data and anticipate flow breakdowns in the future, especially during peak hours. A promising way is to forecast traffic conditions accurately and timely from a short-term perspective and allow traffic managers to understand potential traffic variations instantly. Therefore, as a decision support tool, the short-term traffic flow forecasting model for large-scale traffic data in massive IoT is expected to make a high contribution to active traffic management.

Due to the significance of predicting potential traffic volume in advance, many researchers have devoted themselves to the study of this topic in recent years [4]. Generally, a robust traffic forecast algorithm requires excellent response time and high accuracy. However, the explosive growth of data size makes it difficult to forecast the expected volume timely. Moreover, the prediction is generally based on sampled data with small scales, which decrease the prediction precision to some extent. As the inherent ills of the data-driven traffic forecasting approach, these problems have become a major obstacle to enhance the effectiveness of large-scale traffic management.

In light of the issues above, we propose a 6G-enabled short-term traffic flow forecasting algorithm in a large-scale traffic environment based on time-aware Locality-Sensitive Hashing (LSH) technology, named $TracFore_{time-LSH}$. LSH technology is a fast nearest-neighbor search technology for massive high-dimensional data, which identifies whether the data points are neighbors by mapping them into some buckets. Traditional LSH is usually applied to privacy protection issues in service recommendation scenarios [5] [6] [7]. Furthermore, our $TracFore_{time-LSH}$ is a data-driven prediction approach implemented on real historical sensor data, where the traffic pattern of each sensor is aggregated in 15-min intervals and traffic data transmission between distributed sensors and centralized cloud computing platform is guaranteed by 6G

F. Wang is with School of Computer Science, Qufu Normal University, China. (email: fanwang1997@gmail.com)

M. Zhu is with Facility Horticulture Laboratory of Universities in Shandong, WeiFang University of Science and Technology, ShouGuang, China. (email: zhumin@wfust.edu.cn)

M. Wang is with School of Cyber Science and Engineering, Qufu Normal University, China. (email: wangml@qfnu.edu.cn) [corresponding author]

M. R. Khosravi is with Department of Computer Engineering, Persian Gulf University, Bushehr 7516913817, Iran, and Department of Electrical and Electronic Engineering, Shiraz University of Technology, Shiraz 71557-13876, Iran. (email: mohammadkhosravi@acm.org)

Q. Ni is with School of Computing and Communications, Lancaster University, UK. (email: q.ni@lancaster.ac.uk)

S. Yu is with Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. (email: Shui.Yu@uts.edu.au)

L. Qi is with School of Computer Science and Engineering, Qufu Normal University, China. (email: lianyongqi@qfnu.edu.cn)

technology. In summary, we make the following contributions in this paper.

(1) We propose a novel short-term traffic flow forecasting model based on time-aware Locality-Sensitive Hashing to pursue a more accurate real-time prediction in massive IoT. To the best of our knowledge, this is the first work that incorporates time-aware LSH technology into large-scale traffic forecasting.

(2) We conduct a wide range of experiments based on a large scale real-world Intelligent Transportation Systems (ITS) dataset collected from Nanjing city of China to validate the performance of our proposal. The experimental results show that our $TracFore_{time-LSH}$ outperforms the other two approaches in terms of response time and forecast accuracy.

The rest of the paper is organized as follows. We review related work following this introductory section. Following the related work, the motivation of our research is presented. This is followed by a detailed discussion about how our $TracFore_{time-LSH}$ takes effect as well as the corresponding experimental results. In the last section, we conclude the whole paper and indicate some potential directions in our future work.

II. Related work

Nowadays, many researchers are devoting themselves to technologies in the context of the Internet of Things (IoT) [8] [9] [10]. Based on IoT, Intelligent Transportation Systems (ITS) is emerged as a novel paradigm to manage urban traffic and bring convenience to the lives of residents [11]. As a vital element of ITS, short-term traffic flow forecasting is a crucial topic that forecasts traffic patterns over a few seconds to a few hours. As classified in [12], the traffic flow forecasting approaches can be divided into three categories: naive, parametric, and non-parametric methods. Considering the diversity of short-term traffic flow prediction conditions, we will discuss these three categories in detail according to different traffic contexts.

Naive methods denote the traffic forecasting models based on mathematical statistics, e.g., historical average and clustering approaches. Although Naive methods are with simplicity and efficiency characteristics, they cannot reflect the uncertainty and nonlinearity of traffic dynamics.

Parametric methods utilize the overall distribution of data to estimate a set of parameter values and forecast future traffic patterns. Some typical methods include ARIMA as well as its variation SARIMA based on time series analysis [13], macroscopic traffic flow analysis model for better accuracy [14] to name just a few. Although this kind of methods are with high prediction accuracy, they have a complicated parameter estimation process and have been proven to be unfriendly to unstable traffic environments.

Most of the non-parametric methods are data-driven and free of restriction regarding the data distribution, including neural networks, pattern recognition methods, and so on. In recent years, due to the characteristics of adaptive ability and flexibility, neural networks have received extensive attention from scholars [15] [16]. Li et al. [17] utilize bayesian networks to implement multiple measures chaotic time series prediction approach. Besides, the recurrent neural network (RNN) is very aggressive in processing time series corresponding to traffic patterns, but it is prone to vanishing gradient problems. In this situation, the variants of RNN, long short-term memory (LSTM), and Gated recurrent units (GRU) can better alleviate the issue [18]. Dai et al. [19] develop a gated recurrent units (GRU) model based on traffic information to predict traffic flow in short-term. Ma et al. [20] propose an LSTM model for predicting the time cost during travel in urban. However, the above studies only take time series instead of more comprehensive contexts into account. To overcome their drawbacks, Zhang et al. [21] employ convolutional neural networks (CNN) to combine time and space information to analyze traffic flow data. Nevertheless, all the above neural network methods suffer from common shortcomings that are without high interpretability and really depend on data scale. On the other hand, K-nearest neighbor (K-NN) methods conduct short-term traffic flow forecasting by extracting valuable characteristics in the dataset. Thus, we can draw an understanding of the prediction results from the execution of K-NN. For instance, Lin et al. [22] combine K-NN with local linear wavelet neural network to predict short term traffic flow. Zhang et al. [23] propose an improved K-NN for short-term traffic flow prediction. However, the data-driven K-NN technology consumes a lot of time and its precision is not high enough. Although researchers have made different enhancements on the basis of K-NN, the accuracy of the enhanced K-NN has not been greatly improved and the time consumption continued to increase.

In general, since existing researches are often conducted in various contexts, it is difficult to define whether a method is the best. However, compared with parametric methods, a large number of researchers have concluded that non-parametric methods are better because of their powerful self-learning functions and adaptive capabilities. Thus, we also propose a data-driven non-parametric method, which can achieve more accurate training results in a fairly short response time. Our experimental results demonstrate that our proposal can be easily incorporated into an online traffic control system and achieve better performance.

III. Motivation

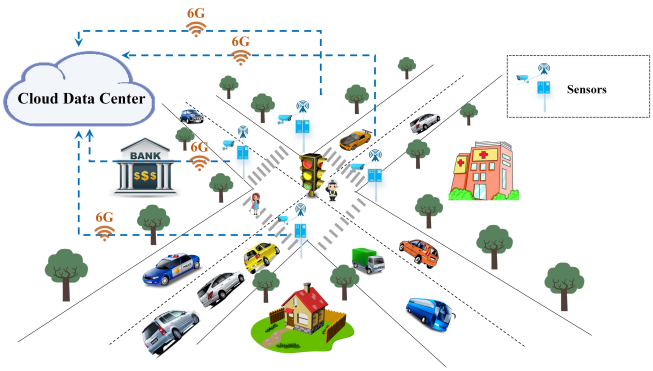


Fig. 1. Traffic dynamics: an example.

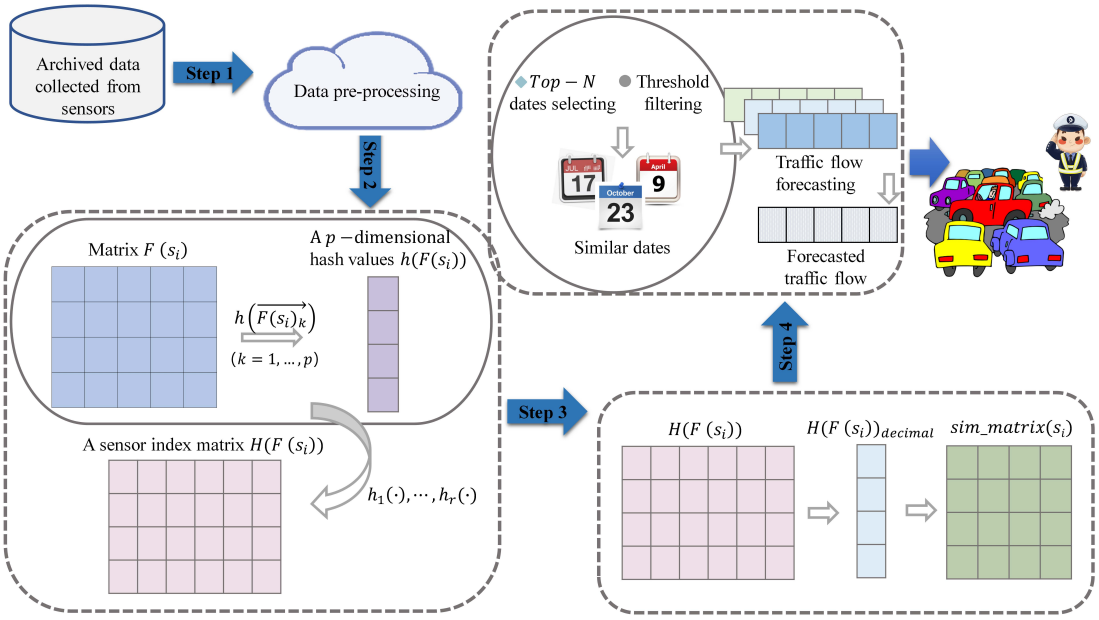


Fig. 3. The technical architecture of our $TracFore_{time-LSH}$.

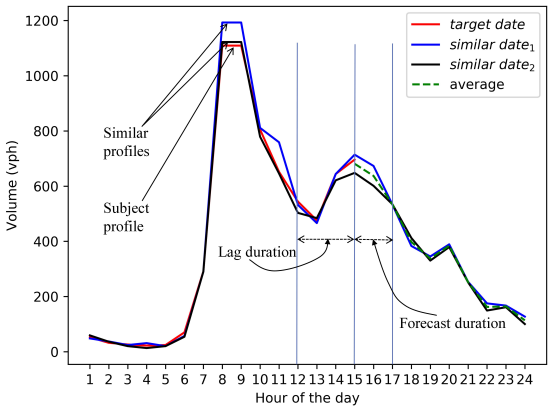


Fig. 2. Graphical representation of our $TracFore_{time-LSH}$.

We employ Fig. 1 to illustrate the motivation of this paper vividly. As shown in Fig. 1, queues and increasingly frequent congestions nowadays require more extensive traffic monitoring. Therefore, the corresponding departments have installed a great deal of devices on the traffic networks, such as sensors in Fig. 1. Based on these sensors, all the real-time traffic data will be provided to the cloud for processing, during which the advanced 6G technology guarantees the efficiency, stability, and integrity of the extensive distributed data transmission. It can be said that implementing 6G-enabled short-term traffic flow forecasting based on the integrated data collected from all sensors is a promising way to provide traffic managers with strategies to anticipate flow breakdowns in the future. However, two issues arise in the traditional short-term traffic flow forecasting methods: (1) The continuous sensors as well as their observed big traffic data render the instant response to variations in traffic conditions infeasible. (2) Only a small portion of sampled data is utilized for traffic flow prediction, which causes the predictive result not accurate enough. Generally, a more effective road capacity management strategy adopted from the forecasting algorithm requires shorter response time and higher accuracy. In light of this situation, we propose an efficient and accurate algorithm named $TracFore_{time-LSH}$, which will be introduced in the subsequent section.

IV. Traffic Flow Forecasting Based on LSH: $TracFore_{time-LSH}$

In this paper, we perform 6G-enabled traffic flow forecasting based on similar flow rate sequences in historical traffic patterns recorded by sensors, where the historical traffic patterns are the search spaces from which we obtain valuable information. Our algorithm is based on the hypothesis that if a previous profile is similar to the current profile, then the subsequent values of the previous profile is similar to the future values of the target profile. Hence, as graphically shown in Fig. 2, given an incomplete traffic sequence of the target day (depicted in solid red line in Fig. 2), i.e., the subject profile desired to be forecasted, our algorithm aims to recognize similar neighbors (depicted in solid black line and blue line in Fig. 2) for it accurately and efficiently from a pool of archived datasets. Concretely, we exploit the sequences in a time window (denoted as lag duration) to determine similar candidates. Then we aggregate the flow rate of the similar profiles in some form and draw the future traffic volume of the subject profile (depicted in broken green line in Fig. 2).

To facilitate the discussion of our proposed $TracFore_{time-LSH}$, we define several symbols as below:

(1) $S = \{s_1, \ldots, s_m\}$: the set of sensors that record the traffic dynamics.

(2) $D = \{d_1, \ldots, d_p\}$: the set of dates in the archived datasets that sensors monitor traffic dynamics.

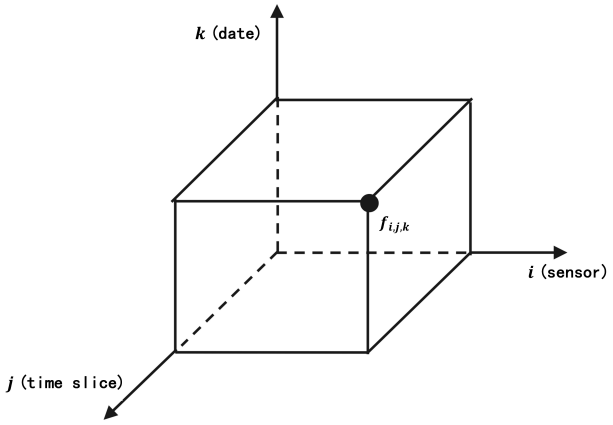


Fig. 4. Traffic flow representation in three-dimensional space.

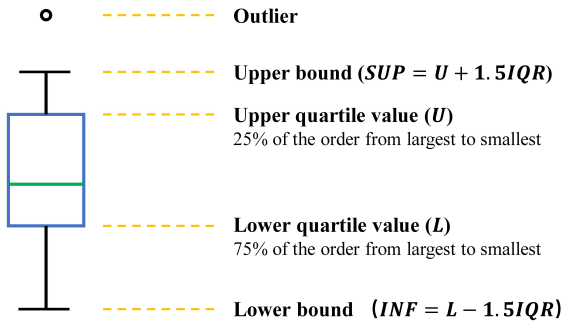


Fig. 5. The composition of classical boxplots.

(3) $T = \{t_1, \ldots, t_n\}$: the set of time slices in the lag duration with fixed time step, where the size of the set is determined by the number of time steps included in the lag duration, e.g., if the lag duration is 1 h and the time step is 15 minutes, then $n = 4$ (1 [h] * 60 [min/h] / 15 [min]).

(4) $f_{i,j,k}$: the traffic flow of the sensor s_i $(1 \leq i \leq m)$ in the j_{th} time slice t_j $(1 \leq j \leq n)$ of the k_{th} day d_k $(1 \leq k \leq p)$.

Then, we will introduce our accurate traffic flow forecasting approach with quick response time based on time-aware LSH, named $TracFore_{time-LSH}$. Fig. 3 shows the technical framework of our methods with 4 steps.

A. Step-1: Data formalization and preprocessing

1) Data formalization: As shown in Fig. 4, the archived traffic profile can be visualized as a three-dimensional space consists of sensor (i), time slice (j), and date (k), where $f_{i,j,k}$ is a point representing the traffic flow in a specific space-time. In this situation, we aggregate the volume of traffic for sensor s_i every 15 minutes (i.e., a time slice) and formalize it as a matrix specified in (1). In this matrix, each row represents the flow of n time slices observed by s_i at a specific date, and each column represents the flow of a certain time slice observed by s_i in p days. It is worth noting that we only utilize the traffic flow of time slices in the lag duration with 15-min intervals to construct the matrix in (1) and perform index table generation as well as similar dates determination subsequently. Here, the number of columns in (1) is the time steps included in the lag duration, i.e., n.

$$F(s_i) = \begin{bmatrix} f_{i,1,1} & \cdots & f_{i,n,1} \\ \vdots & \ddots & \vdots \\ f_{i,1,p} & \cdots & f_{i,n,p} \end{bmatrix} \tag{1}$$

2) data preprocessing: Inevitably, there is some noise in the dataset that harms similar profile recognition and thus results in a bad prediction. To dampen the effect of noise, we first take advantage of boxplots to identify outliers, and then apply winsorization on the abnormal data. Boxplot is a statistical chart based on distance measurement that shows a set of data dispersion. Both [24] and [25] proposed functional boxplots methodologies as informative exploratory tool for outlier detection. Inspired by them, we also employ boxplots to provide a global analysis suitable for the whole data and conduct outlier identification. Fig. 5 introduces the components of the classical boxplots.

Specifically, we perform outlier processing on each row and column of the matrix in (1). In order to ensure the conciseness of this paper, we only introduce the detailed processing of one row of the matrix. Assuming the k_{th} row of the matrix is given by $\overrightarrow{F(s_i)_k}$, where $\overrightarrow{F(s_i)_k} = (f_{i,1,k}, \ldots, f_{i,n,k})$, we first arrange the values of $\overrightarrow{F(s_i)_k}$ in descending order and denote it as $\overrightarrow{c}$, where $\overrightarrow{c} = (c_{i,1,k}, \ldots, c_{i,n,k})$. As shown in Fig. 5, the upper border of the box (enveloped by a blue line) indicates upper quartile value U which is the value at the 25% position of the vector $\overrightarrow{c}$, i.e., $U = c_{i,\frac{n}{4},k}$. Likewise, the lower border of the box indicates the lower quartile value L which is the value at the 75% position of the vector $\overrightarrow{c}$, i.e., $L = c_{i,\frac{3n}{4},k}$. Then the difference between the upper quartile U and the lower quartile L is defined as IQR (inter-quartile range) to represent the 50% central region of the curves, i.e., $IQR = U - L = c_{i,\frac{n}{4},k} - c_{i,\frac{3n}{4},k}$. Actually, IQR is a robust expression of data characteristics because it covers the 50% central range of the data, which will not be affected by outliers. The whisker of the boxplot is the black vertical lines extending from the edge of the box in Fig. 5, which indicates the maximum range of data except for outliers. Now, we begin to detect outliers. We first extend the range of the 50% central range by 1.5 times to obtain the upper and lower bounds of the data. Formally, we define the upper bound as SUP, where $SUP = U + 1.5IQR$, and the lower bound as INF, where $INF = L - 1.5IQR$. We regard the points outside these two bounds as potential outliers. Here, the coefficient 1.5 is suggested by [24] as well as [25], and can be proved by the standard normal distribution. The reader can refer to these two researches for more detailed discussion. Afterwards, add data points larger than the upper bound SUP to the point set SOt, where $SOt = \{c_{i,j,k} | c_{i,j,k} \geq SUP\}$, and add data points smaller than the lower bound INF to the point set IOt, where $IOt = \{c_{i,j,k} | c_{i,j,k} \leq INF\}$. In this situation, $SOt \cup IOt$ is the set of all outliers. Finally, we use Eq. (2) to perform winsorization on identified outliers by replacing

the abnormal data points with the closest values in the normal range.

$$\begin{cases} c_{i,j,k}=c_{i,\frac{n}{4},k}+1.5(c_{i,\frac{n}{4},k}-c_{i,\frac{3n}{4},k}) & \forall c_{i,j,k} \in SOt \\ c_{i,j,k}=c_{i,\frac{3n}{4},k}-1.5(c_{i,\frac{n}{4},k}-c_{i,\frac{3n}{4},k}) & \forall c_{i,j,k} \in IOt \end{cases} \quad (2)$$

B. Step-2: Building a sensor index table.

In this subsection, we focus on how to build a time-aware sensor index table based on LSH by using the matrix in (1). Generally, the *cosine distance* is of great significance in spaces that have multi-dimensions. Due to that vehicles passing through the sensor s_i in different time slices may construct a multi-dimensional vector, thus we utilize time-aware LSH technology corresponding to the cosine distance to achieve similarity computation between traffic profiles of different days. Concretely, for the k_{th} row $\overrightarrow{F(s_i)_k}$ of matrix $F(s_i)$, where $\overrightarrow{F(s_i)_k} = (f_{i,1,k}, \ldots, f_{i,n,k})$, we first transform it into a hash value $h(\overrightarrow{F(s_i)_k})$ using the LSH function in (3). Here, $\overrightarrow{v}$ is an n-dimensional vector $(v_1, \ldots, v_n)$ that randomly generated in the space of [-1, 1], where v_j is a random value in the range [-1, 1]; The symbol $\cdot$ denotes the dot product operation of two vectors.

$$h(\overrightarrow{F(s_i)_k}) = \begin{cases} 1 & if\ \overrightarrow{F(s_i)_k} \cdot \overrightarrow{v} > 0 \\ 0 & if\ \overrightarrow{F(s_i)_k} \cdot \overrightarrow{v} \leq 0 \end{cases} \quad (3)$$

After performing the hash mapping in (3) on the k_{th} row of matrix $F(s_i)$, the row vector representing the traffic flow on the k_{th} day is mapped to a Boolean value. Repeat this process for each row in (1) until all rows are mapped, i.e., a p-dimensional Boolean vector $h(F(s_i))$ is obtained in (4).

$$h(F(s_i)) = (h(\overrightarrow{F(s_i)_1}), \ldots, h(\overrightarrow{F(s_i)_p})^T \quad (4)$$

Through the above process, the traffic characteristics of each date in the matrix $F(s_i)$ will be transformed into a unique Boolean value. However, LSH is a probability-based similar candidate identification technique, and hash values mapped by only one hash function in (3) can't guarantee an accurate expression of the traffic characteristics. To address this issue, hash functions $h_1(\cdot), \ldots, h_r(\cdot)$ randomly generated by (2) are employed to achieve r transformations from $F(s_i)$ in (1) to $h(F(s_i))$ in (4). Now, we can obtain a $p * r$ Boolean matrix $H(F(s_i))$ in (5), i.e., the time-aware sensor index reflecting the traffic pattern of s_i.

$$H(F(s_i)) = \begin{bmatrix} h_1(\overrightarrow{F(s_i)_1}) & \cdots & h_r(\overrightarrow{F(s_i)_1}) \\ \vdots & \ddots & \vdots \\ h_1(\overrightarrow{F(s_i)_p}) & \cdots & h_r(\overrightarrow{F(s_i)_p}) \end{bmatrix} \quad (5)$$

Repeat the above process for each sensor in set S to build its time-aware index matrix $H(F(s_i))$ in (5), and we can finally obtain a sensor index table denoted as $Table_{index}$, which contains traffic characteristics of all sensors.

Algorithm 1: $TracFore_{time-LSH}$

Input:
s_{target}: the target sensor
$S = \{s_1, \ldots, s_m\}$: sensor set
$D = \{d_1, \ldots, d_p\}$: date set
$T = \{t_1, \ldots, t_n\}$: time slice set
$f_{i,j,k}$: traffic flow of sensor s_i in time slice t_j of date d_k
Output:
f_{target,J,k_1}: traffic flow of sensor s_i in desired time slice t_J of date d_{k_1}.

for $x = 1$ *to* r **do**
  for $j = 1$ *to* n **do**
    v_j = random [-1,1]
  $h_x(\cdot) = (v_1, \ldots, v_n)$
for *each* $s_i \in S$ **do**
  generate time matrix $F(s_i)$ in Eq.(1)
  preprocess data in $F(s_i)$
  for $k = 1$ *to* p **do**
    $h(\overrightarrow{F(s_i)_k}) = \overrightarrow{F(s_i)_k} * h_x(\cdot)$
  $h(F(s_i)) = (h(\overrightarrow{F(s_i)_1}), \ldots, h(\overrightarrow{F(s_i)_p}))^T$
for *each* $s_i \in S$ **do**
  generate $H(F(s_i))$ using Eq. (5)
generate sensor index table for all sensors
for *each* $s_i \in S$ **do**
  for $k = 1$ *to* p **do**
    decimal conversion from $H(F(s_i))_k$ to $A_k(s_i)$
  if $A_{k_1}(s_i) = A_{k_2}(s_i)$ **then**
    $sim_{k_1,k_2} = 1$
  else
    $sim_{k_1,k_2} = 0$
Generate a hash table H_{table} based on "$s_i \rightarrow sim\_matrix(s_i)$" mappings
Repeat the above process to generate L hash tables $H\_table_1, \ldots, H\_talbe_L$
Set a date d_{k_1} and a time slice t_J in which the traffic flow needs to be predicted
Calculate $SIM\_M(s_{target})$ using Eq.(8)
Select $top-K$ similar dates into $List(d_{k_1})$
Set a similarity threshold
for $k_2 = 1$ *to* p **do**
  if $k_2 \in List(d_{k_1})$, $k_2 \neq k_1$ *and* $sim_{k_1,k_2}(s_{target}) \geq threshold$ **then**
    f_{target,J,k_1} is predicted by Eq.(10)
return f_{target,J,k_1}

C. Step-3: Determination of similar dates

In this subsection, we will define the similarity between different dates of sensors based on the sensor index table $Table_{index}$. Although $Table_{index}$ contains index matrix of all sensors, we only consider the date similarity calculation of sensor s_i as an example. Since each row of matrix $H(F(s_i))$ in

(4) is an r-dimensional 0-1 string, we can regard it as a unique binary value and convert it into a corresponding decimal value. For example, when $r = 4$ and the string of the k_{th} row in matrix $H(F(s_i))$ is "0110", we convert this 4-dimensional 0-1 string into a unique decimal number "6". In this way, we can convert the time-aware sensor index matrix $H(F(s_i))$ into the p-dimensional decimal vector $H(F(s_i))_{decimal}$ in (6), where each decimal value uniquely represents the traffic flow characteristics of the corresponding date.

$$H(F(s_i)_{decimal}) = (A_1(s_i), \ldots, A_p(s_i))^T \tag{6}$$

According to the column vector in (6), we compare the decimal hash values of p dates. When $A_{k_1}(s_i) = A_{k_2}(s_i)$ $(k_1 - k_2 \neq 0)$, we assign $sim_{k_1,k_2}(s_i)$ to 1; otherwise, we assign $sim_{k_1,k_2}(s_i)$ to 0. Here, sim_{k_1,k_2} indicates the similarity between traffic flow patterns of s_i during date d_{k_1} and date d_{k_2}. Specifically, when $d_{k_1} = d_{k_2}$, we assign $sim_{k_1,k_2}(s_i)$ to 0 as well, because the similarity of traffic characteristics on the same day is meaningless. Thus, for the sensor s_i, a p*p-dimensional Boolean matrix is obtained as denoted in (7), which can describe the similarity of the traffic flow characteristics of the sensor s_i on different days. The mapping from s_i to $sim\_matrix(s_i)$ can form a hash table (named as H_{table}), which can be generated offline.

$$sim\_matrix(s_i) = \begin{bmatrix} sim_{1,1}(s_i) & \cdots & sim_{1,p}(s_i) \\ \vdots & \ddots & \vdots \\ sim_{p,1}(s_i) & \cdots & sim_{p,p}(s_i) \end{bmatrix} \tag{7}$$

For sensor s_i, due to that our LSH technology is actually a probability-based similar candidate identification technique, utilizing only one hash table can't guarantee the similarity retention of traffic characteristics on different dates. Thus, we need to generate L hash tables offline (i.e., $H\_table_1, \ldots, H\_talbe_L$) and establish their L similarity matrices for sensor s_i (i.e., $sim\_matrix(s_i)_1, \ldots, sim\_matrix(s_i)_L$). Afterwards, SIM_M is obtained by using Eq. (8) to perform an "AND" operation that accumulates the value of the corresponding position in different similarity matrices, where each entry is in the range [0, L], and a higher similarity value indicates that the corresponding two dates have a high probability of being similar.

$$SIM\_M(s_i) = sim\_matrix(s_i)_1 + \ldots + sim\_matrix(s_i)_L \tag{8}$$

D. Step-4: Traffic flow forecasting

According to the similarity matrix in Step-3, for sensor s_i, we can predict the traffic flow rate in the forecast duration of its date d_{k_1}. First, we add the $top - K$ most similar dates to s_i's neighbor list $List(d_{k_1})$. Afterwards, we set a similarity threshold defined by Eq. (9) to filter out dates whose similarity is less than the threshold. Here, $Sim\_set(d_{k_1})$ in (9) is the set of candidate dates that are actually similar to d_{k_1}. Finally, the traffic profile of dates in $Sim\_set(d_{k_1})$ can provide a reference by Eq. (10) for traffic volume forecasting, where J denotes the time slice when we intend to predict traffic flow rate.

$$Sim\_set(d_{k_1}) = \{d_{k_2} | sim_{k1,k2}(s_i) \geq threshold, d_{k_2} \in List(d_{k_1})\} \tag{9}$$

$$f_{i,J,k_1} = \frac{\sum_{d_{k_2} \in Sim\_set(d_{k_1})} sim_{k_1,k_2}(s_i) f_{i,J,k_2}}{\sum_{d_{k_2} \in Sim\_set(d_{k_1})} sim_{k_1,k_2}(s_i)} \tag{10}$$

In summary, we design the pseudo code in Algorithm 1 to formulate our $TracFore_{time-LSH}$.

V. Experiments

A. dataset and evaluation metrics

In this paper, we employ a real-world dataset collected from Nanjing city of China to conduct our experiments, which was generously provided by Xu et al. [26]. The dataset contains 332 active sensors as well as their recorded traffic information on the streets within Nanjing. According to this dataset, a comprehensive analysis can be made corresponding to different driving environments and the effectiveness of our proposal can be examined under various traffic conditions. Specifically, we aggregate the count of vehicles passing each sensor at 15-min intervals based on the dataset and implements our model.

Furthermore, we examine the performance of the proposed method $TracFore_{time-LSH}$ by metrics shown in Eqs. (11)-(13). Here, MAPE (Mean Absolute Percentage Error) considers the percentage between forecast error and the observed value; MAE (Mean Absolute Error) considers the average deviation between the observed and predicted traffic flow rates; RMSE (Root Mean Square Error) considers the square root of the mean-variance of the difference in the vehicle numbers. In these equations, F_i is the i_{th} predicted value, O_i is the i_{th} real value, N is the size of our samples. According to these matrices, a further and broader evaluation of predicted performance can be conducted.

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{F_i - O_i}{O_i} \right| \times 100\% \tag{11}$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |F_i - O_i| \tag{12}$$

$$RMSE = \frac{1}{N} \sqrt{\frac{\sum_{i=1}^{N} (F_i - O_i)^2}{N}} \tag{13}$$

We will discuss the performance of our proposal from two perspectives: (1) by the level of traffic flow when the forecast occurs; (2) by the time of day when the forecast occurs. Concretely, for (1), we classify the traffic flow in increments of 500 veh/h/ln in different levels and investigate the accuracy of our short-term traffic forecast corresponding to each level. The volume bins with different levels is shown in Table I. For (2),

TABLE I
CLASSIFICATION OF TRAFFIC LEVELS (PRESENTED BY [27])

Volume groups	Group description
Group 1	$\geq$ 0 and $<$ 500 veh/h/ln
Group 2	$\geq$ 500 and $<$ 1000 veh/h/ln
Group 3	$\geq$ 1000 and $<$ 1500 veh/h/ln
Group 4	$\geq$ 1500 and $<$ 2000 veh/h/ln
Group 5	$\geq$ 2000 veh/h/ln

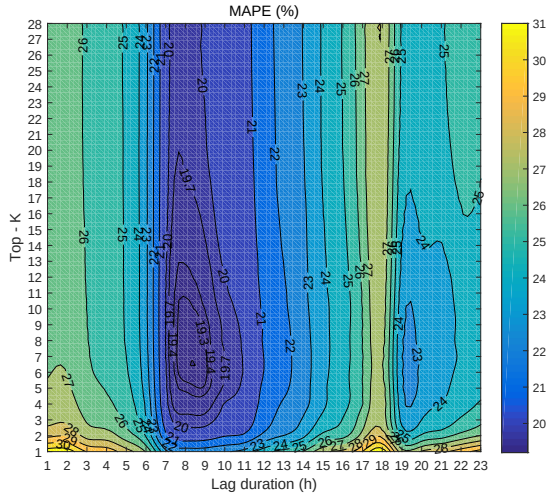


Fig. 6. Forecast errors of different size of lag duration and candidates.

we examine the performance of predictions made at different hours in a day that includes 24 hours. Such evaluations with these two perspectives can provide an in-depth analysis of the method performance.

B. Parameter adjustment for $TracFore_{time-LSH}$

It is of necessity to determine several parameters in advance to ensure that our method can achieve the best performance. The parameters include lag duration (i.e., n), the number of neighbors (i.e., $Top-K$), the number of hash functions (i.e., r) and the number of hash tables (i.e., L).

1) identifying optimal lag duration and candidate number: Lag duration and candidates play a significant role in our short-term traffic forecast algorithm, and the size of them determines whether similar dates can be identified accurately. Fig. 6 depicts the effect of different sizes of lag duration and candidates in terms of the performance indicator representing by MAPE. In Fig. 6, we can observe that the 8-hour lag duration is most suitable for the context of our $TracFore_{time-LSH}$. Similarly, our short-term traffic flow forecasting generally performs better when the number of candidates is around 7. According to the above analysis, we set n as 8 and $top-K$ as 7 to achieve similar dates search technology and traffic prediction algorithm effectively and accurately.

2) identifying optimal number of hash functions and hash tables: Due to that LSH is a probability-based similar candidate identification technique, the optimal number of hash functions and hash tables ensures the stability of our forecast.

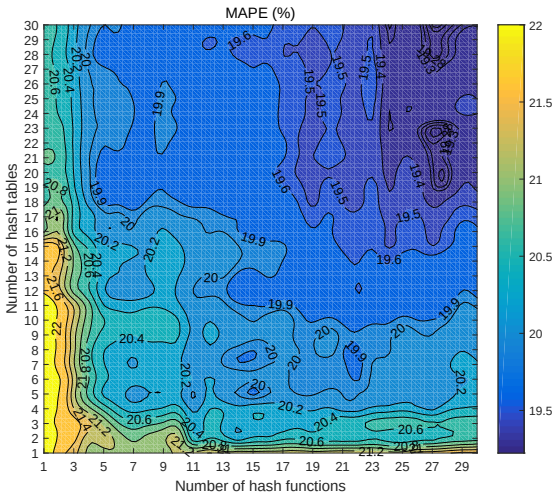


Fig. 7. Forecast errors of different number of hash functions and hash tables.

We examine the impact of the number of hash functions and hash tables on forecast accuracy in terms of MAPE, as intuitively shown in Fig. 7. It can be observed that as the number of hash functions and hash tables increases, forecast accuracy increases. This result occurs because more hash functions indicate a stricter condition for similar date searches, while more hash tables indicate a more stable performance. Specifically, we depict the contours of the best-performing areas in Fig. 7 with steps of 0.02. According to Fig. 7, we can conclude that our $TracFore_{time-LSH}$ performs well when $r = 27$ and $L = 30$.

In addition, we set the threshold in (9) to 1 to filter the dates in neighbor list $List(d_{k_1})$ that are actually dissimilar to d_{k_1}.

C. Predictive accuracy by the level of traffic and time of day

In this subsection, we examine the accuracy of the proposed short-term traffic flow forecasting approach in terms of the level of traffic and time of day. The experimental results are shown in Fig. 8. Intuitively, we employ boxplots (as introduced in Fig. 5) to illustrate the spread of the prediction errors, where the blue solid line denotes the average effect of our algorithm.

As shown in Fig. 8, when we examine the forecast errors in terms of the level of traffic, especially for MAPE, a corresponding reduction in error is found as the flow level increases. Generally, MAPE provides a better perspective in measuring traffic forecast accuracy, which is because MAPE normalizes errors by considering the percentage between forecast error and the observed value. However, MAE and RMSE have opposite performances to MAPE as traffic levels increase, which is because MAE and RMSE only consider the absolute deviation between the forecast value and the observed value.

As expected, the same things occur when we test the forecast errors by the time of day. For MAPE, our $TracFore_{time-LSH}$ performs better during peak-hours than during off-peak hours. However, MAE and RMSE have smaller errors for time with fewer vehicles (e.g., at midnight) because of the low observed volume.

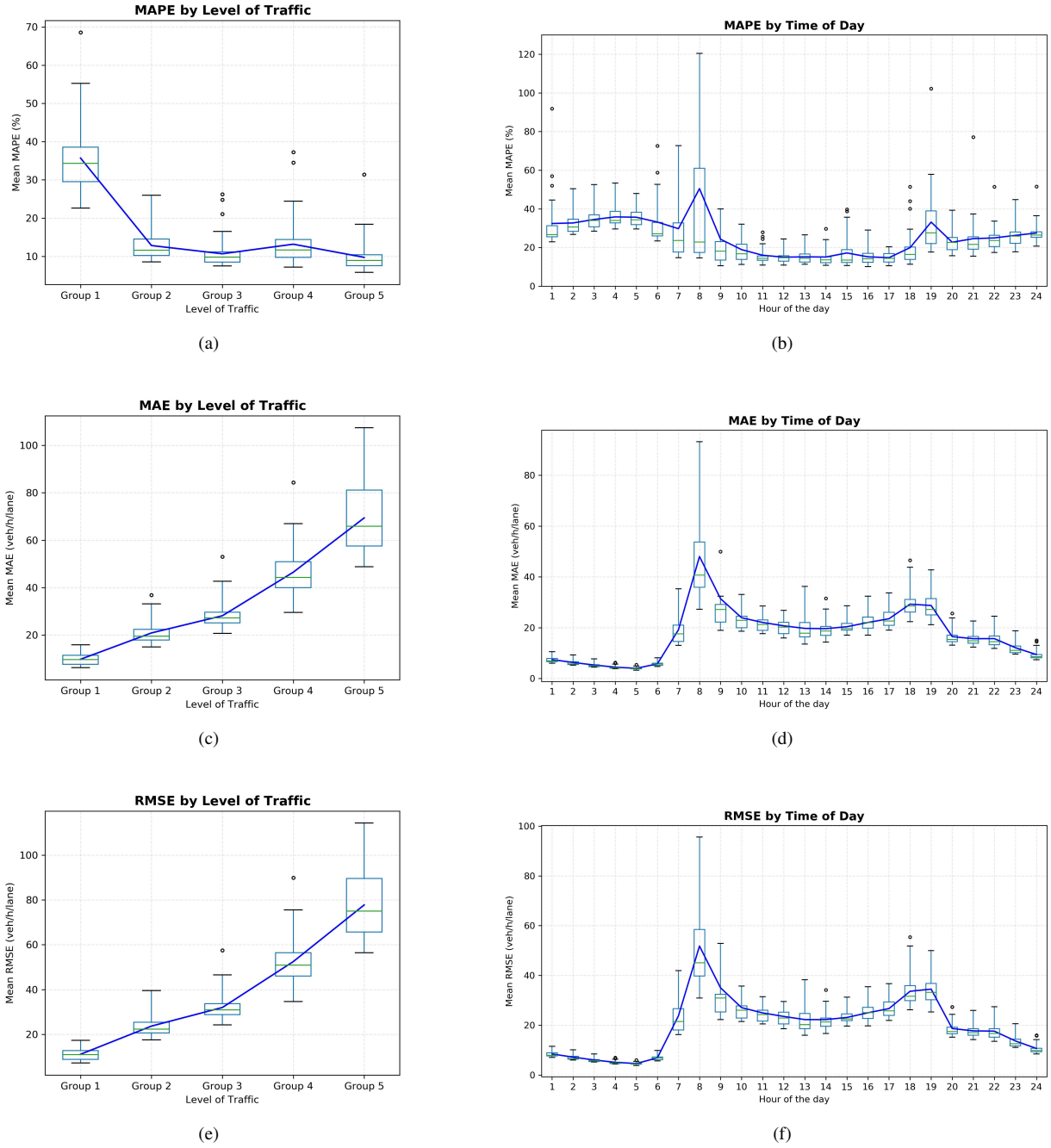


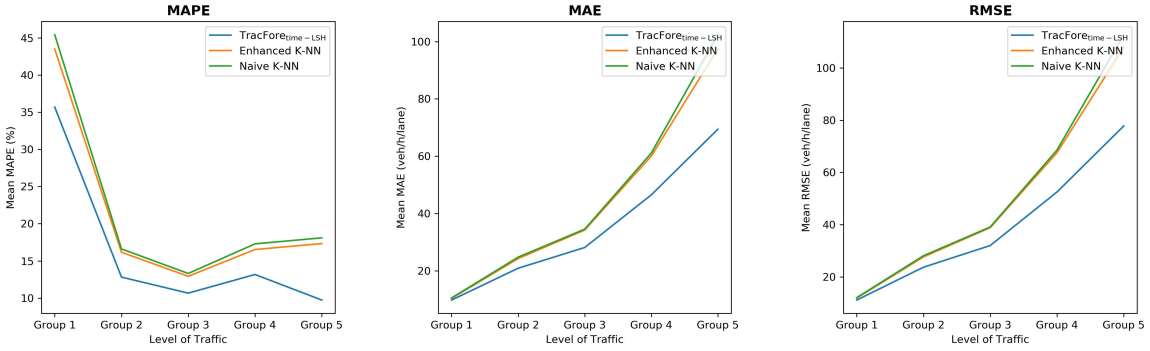
Fig. 8. Forecast errors of $TracFore_{time-LSH}$ by the level of traffic and time of the day.

Generally, according to the whisker distribution in Fig. 8, it is evident that our $TracFore_{time-LSH}$ can provide a reliable and accurate prediction result, especially for high-level traffic (which always occurs during peak hours). In the real world, congestions with negative effects occur increasing frequently, thus it is becoming a necessity to forecast the traffic flow during peak hours. Consequently, it is a promising way to apply our stable and effective traffic forecasting method $TracFore_{time-LSH}$ to understand congested traffic conditions and anticipate flow breakdowns in the future.

D. Experimental comparison results and analysis

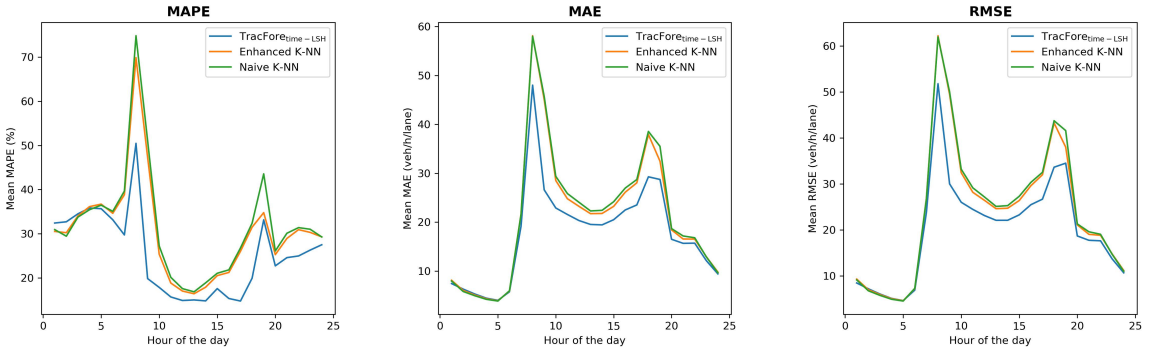
To verify the performance of our proposal, we compare $TracFore_{time-LSH}$ with two state-of-the-art methods: Naive K-NN and Enhanced K-NN [28]. We employ Fig. 9 and Fig. 10 to illustrate the experimental contrast effect in terms of the level of traffic and time of the day (The reason for the variation trend of the curves in Fig. 9 and Fig. 10 is the same as that in Fig. 8, and will not be repeated here). Evidently, our $TracFore_{time-LSH}$ can provide a lower forecast error in short-term traffic flow prediction, especially during high traffic levels and peak hours.

To test these three approaches thoroughly on short-term traf-



(a) (b) (c)

Fig. 9. Forecast errors compared with the other two approaches by level of traffic.



(a) (b) (c)

Fig. 10. Forecast errors compared with the other two approaches by time of the day.

fic flow forecast errors, we compare their average performance in Table II. The experimental results show that relative to the average accuracy of Naive K-NN and Enhanced K-NN, our $TracFore_{time-LSH}$ is reduced by 32.6% for MAPE, 7.5% for MAE and 14.6% for RMSE. Consequently, our proposal outperforms the other two methods in terms of short-term flow forecasts. This result occurs because we not only filter noise in the dataset with effective data preprocessing technology, but also creatively apply time-aware LSH in finding the most similar dates to the target date. In this way, these so picked neighbors are employed to perform traffic flow prediction based on the nature of LSH.

Generally, the efficiency of an algorithm is also a significant indicator to measure its ability, which is especially important for the real-time pre-perception of traffic dynamics. Thus, we record the average time cost of the three methods (i.e., $TracFore_{time-LSH}$, Enhanced K-NN, and Naive K-NN), as presented in Table III. The results indicate that the time cost of our algorithm is found to be quite small and remains approximately stable compared with the other two methods. Numerically, the time cost of our $TracFore_{time-LSH}$ is reduced by 97.3% compared with the average time cost of Naive K-NN and Enhanced K-NN. The reason is that most of the work in our proposal (e.g., hash table creation and similarity calculation) can be completed offline and the remaining work (e.g., similar dates search and flow forecasts) can be finished quite efficiently based on the stored information. In contrast, Enhanced K-NN is time-consuming because it takes a lot of time in optimizing its performance. The results in Table III mean that our $TracFore_{time-LSH}$ can usually meet the demand for "immediate response" of traffic flow forecasting, thereby providing more effective road capacity management.

TABLE II
MEAN FORECAST ERRORS OF THE THREE METHODS

model	MAE	RMSE	MAPE
Naive K-NN	22.01	47.91	29.01%
Enhanced K-NN	21.53	45.43	28.07%
TracFore$_{\text{time-LSH}}$	20.14	39.86	19.23%

TABLE III
PREDICTIVE EFFICIENCY OF THE THREE METHODS

model	TracFore$_{\text{time-LSH}}$	Enhanced K-NN	Naive K-NN
time cost (mm)	4.3	298.6	17.4

VI. CONCLUSION

In this paper, we propose $TracFore_{time-LSH}$, a data-driven and non-parametric approach aided by IoT and 6G technologies, which utilizes big traffic data detected from sensors

to perform short-term traffic flow forecasts in massive IoT. The algorithm adopts time-aware Locality Sensitive Hashing for massive high-dimensional traffic data to achieve a timely and accurate prediction. It assists traffic managers in developing proactive traffic management strategies and anticipating flow breakdowns in the future. $TracFore_{time-LSH}$ provides a novel perspective to predict the expected volume and can achieve a good tradeoff between response time and prediction accuracy in large-scale traffic data environment. The experimental results demonstrate the effectiveness and availability of our $TracFore_{time-LSH}$.

In addition to the traffic patterns in the archived data, complex application contexts, e.g., weather, incident, and road work, also play a significant role in prediction performance. One of the limitations of our research is that we fail to take these impactive factors into account for more accurate precision. In future work, we will include more traffic conditions as a valuable supplement to our study. Furthermore, privacy concerns as an important factor in traffic scenes will also be treated in our future research [29] [30].

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (No. 61872219) and the Natural Science Foundation of Shandong Province (ZR2019MF001).

REFERENCES

[1] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, "Variational lstm enhanced anomaly detection for industrial big data," *IEEE Transactions on Industrial Informatics*, doi: 10.1109/TII.2020.3022432, 2020.

[2] X. Li, H. Mengyan, Y. Liu, V. G. Menon, A. Paul, and Z. Ding, "I/q imbalance aware nonlinear wireless-powered relaying of b5g networks: Security and reliability analysis," *IEEE Transactions on Network Science and Engineering*, doi: 10.1109/TNSE.2020.3020950, 2020.

[3] S. Jacob, V. G. Menon, P. G. Shynu, S. K. S. Fathima, B. Mahapatra, and S. Joseph, "Bidirectional multi-tier cognitive swarm drone 5g network," pp. 1219–1224, doi: 10.1109/INFOCOMWKSHPS50562.2020.9162676, 2020.

[4] X. Xu, Z. Fang, L. Qi, X. Zhang, Q. He, and X. Zhou, "Tripres: Traffic flow prediction driven resource reservation for multimedia iov with edge computing," *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2020.

[5] X. Chi, C. Yan, H. Wang, W. Rafique, and L. Qi, "Amplified locality-sensitive hashing-based recommender systems with privacy protection," *Concurrency and Computation: Practice and Experience*, doi: 10.1002/cpe.5681, 2020.

[6] L. Wang, X. Zhang, R. Wang, C. Yan, H. Kou, and L. Qi, "Diversified service recommendation with high accuracy and efficiency," *Knowledge-Based Systems*, vol. 204, p. 106196, 2020.

[7] H. Liu, H. Kou, C. Yan, and L. Qi, "Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph," *Complexity*, vol. 2020, pp. 1–15, 2020.

[8] X. Xu, Q. Huang, X. Yin, M. Abbasi, M. Khosravi, and L. Qi, "Intelligent offloading for collaborative smart city services in edge computing," *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2020.3000871, 2020.

[9] X. Zhou, W. Liang, K. I. Wang, H. Wang, L. T. Yang, and Q. Jin, "Deep-learning-enhanced human activity recognition for internet of healthcare things," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6429–6438, 2020.

[10] H. Zhang, M. Babar, M. U. Tariq, M. A. Jan, V. G. Menon, and X. Li, "Safecity: Toward safe and secured data management design for iot-enabled smart city planning," *IEEE Access*, vol. 8, pp. 145 256–145 267, 2020.

[11] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Bhuiyan, "Adaptive computation offloading with edge for 5g-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, doi: 10.1109/TITS.2020.2982186, 2020.

[12] E. Kechagias, S. Gayialis, G. D. Konstantakopoulos, and G. Papadopoulos, "Traffic flow forecasting for city logistics: a literature review and evaluation," *International Journal of Decision Support Systems*, vol. 4, p. 159, 2019.

[13] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[14] J. Xiao and Z. Wang, "Traffic speed cloud maps: A new method for analyzing macroscopic traffic flow," *Physica A: Statistical Mechanics and its Applications*, vol. 508, pp. 367–375, 2018.

[15] X. Zhou, Y. Li, and W. Liang, "Cnn-rnn based intelligent recommendation for online medical pre-diagnosis support," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, doi: 10.1109/TCBB.2020.2994780, 2020.

[16] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. Zhao, "Exploiting multi-channels deep convolutional neural networks for multivariate time series classification," *Frontiers of Computer Science*, vol. 10, pp. 96–112, 2016.

[17] Y. Li, X. Jiang, H. Zhu, S. Peeta, X. S. He, T. Zheng, and Y. Li, "Multiple measures based chaotic time series for traffic flow prediction based on bayesian theory," *Nonlinear Dynamics*, vol. 85, pp. 179–194, 2016.

[18] L. Li, "Online short-term traffic flow prediction considering the impact of temporal-spatial features," *Journal of Transportation Systems Engineering and Information Technology*, vol. 16, pp. 165–171, 2016.

[19] 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. 143 025–143 035, 2019.

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

[21] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, "Short-term traffic flow prediction based on spatio-temporal analysis and cnn deep learning," *Transportmetrica A: Transport Science*, vol. 15, no. 2, pp. 1688–1711, 2019.

[22] L. Lin, Y. Li, and A. Sadek, "A k nearest neighbor based local linear wavelet neural network model for on-line short-term traffic volume prediction," *Procedia-Social and Behavioral Sciences*, vol. 96, pp. 2066–2077, 2013.

[23] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, "An improved k-nearest neighbor model for short-term traffic flow prediction," *Procedia-Social and Behavioral Sciences*, vol. 96, pp. 653–662, 2013.

[24] Y. Sun and M. G. Genton, "Functional boxplots," *Journal of Computational and Graphical Statistics*, vol. 20, pp. 316–334, 2011.

[25] Y. Sun and M. G. Genton, "Adjusted functional boxplots for spatio-temporal data visualization and outlier detection," *Environmetrics*, vol. 23, pp. 54–64, 2012.

[26] X. Xu, B. Shen, X. Yin, M. Khosravi, H. Wu, L. Qi, and S. Wan, "Edge server quantification and placement for offloading social media services in industrial cognitive iov," *IEEE Transactions on Industrial Informatics*, doi: 10.1109/TII.2020.2987994, 2020.

[27] J. Guo, W. Huang, and B. Williams, "Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50–64, 2014.

[28] F. G. Habtemichael and M. Cetin, "Short-term traffic flow rate forecasting based on identifying similar traffic patterns," *Transportation Research Part C: Emerging Technologies*, vol. 66, pp. 61–78, 2016.

[29] L. Qi, C. Hu, X. Zhang, M. Khosravi, S. Sharma, S. Pang, and T. Wang, "Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment," *IEEE Transactions on Industrial Informatics*, doi: 10.1109/TII.2020.3012157, 2020.

[30] W. Zhong, X. Yin, X. Zhang, S. Li, W. Dou, R. Wang, and L. Qi, "Multi-dimensional quality-driven service recommendation with privacy-preservation in mobile edge environment," *Computer Communications*, vol. 157, pp. 116–123, 2020.

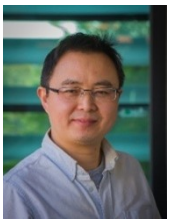
Fan Wang Fan Wang received her bachelor degree in 2019 from the Department of Mathematics, Qufu Normal University, China. Now, she is pursuing her master degree in the School of Computer Science, Qufu Normal University, China. Her research interests include big data analyses and recommender systems.

Zhu Min Zhu Min, born in 1984 in Shouguang, China, graduated from Liaocheng University in 2008, majoring in electronic information science and technology. Since graduation, she has been working in the forefront of education, and obtained a master's degree in Electronic and Communication Engineering from Ocean University of China. Now, she is a lecturer of WeiFang University of Science and Technology, Shouguang, China. Her research interests include data analyses and intelligent information systems.

Maoli Wang Maoli Wang received the B.S. degree in automation from Qufu Normal University, China, in 2004, and the M.S. degree and the Ph.D degree in control theory and control engineering from Harbin Engineer University, China, in 2008. He is a professor at School of Cyber Science and Engineering, Qufu Normal University, Rizhao, China. His research interests include the internet of things, blockchain, and artificial intelligence.

Mohammad R. Khosravi Mohammad R. Khosravi is now with the Department of Computer Engineering, Persian Gulf University, Bushehr, Iran, and has been with Department of Electrical and Electronic Engineering, Shiraz University of Technology, Shiraz, Iran. Mohammad has studied electrical engineering with expertise in communications and signal processing for BSc, MSc and PhD degrees, all from Iranian universities in 2013, 2015 and 2020, respectively. His main interests include statistical signal and image processing, medical bioinformatics, radar imaging and satellite remote sensing, and computer communications.

Qiang Ni Qiang Ni (M'04-SM'08) received the B.Sc., M.Sc., and Ph.D. degrees from the Huazhong University of Science and Technology, China, all in engineering. He is a Professor and the Head of Communication Systems Research Group, School of Computing and Communications, Lancaster University, InfoLab21, Lancaster, U.K. His research interests include future generation communications and networking systems, including green communications and networking, cloud systems, cognitive radio network systems, heterogeneous networks, 5G, SDN, IoTs, big data analytics and vehicular networks in which areas he had already published more than 180 papers. He is a Voting Member of IEEE 1932.1 standard. He was an IEEE 802.11 Wireless Standard Working Group Voting member and a Contributor to the IEEE Wireless Standards.



Shui Yu Shui Yu (Senior Member, IEEE) is currently a Professor with the School of Computer Science, University of Technology Sydney, Australia. He has published two monographs and edited two books and more than 300 technical articles, including top journals and top conferences, such as the IEEE TPDS, TC, TIFS, TMC, TKDE, TETC, ToN, and INFOCOM. His research interests include security and privacy, networking, big data, and mathematical modeling. He initiated the research field of networking for big data in 2013. His h-index is 41. He is a member of AAAS and ACM. He is currently serving a number of prestigious editorial boards, including the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS (Area Editor) and the IEEE Communications Magazine. He is a Distinguished Lecturer of the IEEE Communication Society.

Lianyong Qi Lianyong Qi received his PhD degree in Department of Computer Science and Technology from Nanjing University, China, in 2011. He is a professor of Qufu Normal University, China. He has published over 90+ peer-reviewed journal and conference papers (first author or corresponding author). His research interests include recommender systems and services computing. He is the managing editor of Journal of Organizational and End User Computing.