



A Survey of Traffic Prediction: from Spatio-Temporal Data to Intelligent Transportation

Haitao Yuan¹ · Guoliang Li¹

Received: 29 July 2020 / Revised: 27 December 2020 / Accepted: 28 December 2020 / Published online: 23 January 2021
© The Author(s) 2021

Abstract

Intelligent transportation (e.g., intelligent traffic light) makes our travel more convenient and efficient. With the development of mobile Internet and position technologies, it is reasonable to collect spatio-temporal data and then leverage these data to achieve the goal of intelligent transportation, and here, traffic prediction plays an important role. In this paper, we provide a comprehensive survey on traffic prediction, which is from the spatio-temporal data layer to the intelligent transportation application layer. At first, we split the whole research scope into four parts from bottom to up, where the four parts are, respectively, spatio-temporal data, preprocessing, traffic prediction and traffic application. Later, we review existing work on the four parts. First, we summarize traffic data into five types according to their difference on spatial and temporal dimensions. Second, we focus on four significant data preprocessing techniques: map-matching, data cleaning, data storage and data compression. Third, we focus on three kinds of traffic prediction problems (i.e., classification, generation and estimation/forecasting). In particular, we summarize the challenges and discuss how existing methods address these challenges. Fourth, we list five typical traffic applications. Lastly, we provide emerging research challenges and opportunities. We believe that the survey can help the partitioners to understand existing traffic prediction problems and methods, which can further encourage them to solve their intelligent transportation applications.

Keywords Traffic prediction · Spatio-temporal data · Intelligent transportation

1 Introduction

With the development of Internet and position technologies, there are more and more spatio-temporal data collected by governments or some transportation companies. For example, Didi¹ and Uber², respectively, handle 30 million and 18 million ride orders per day, and they would collect corresponding trajectories if these orders are finished. It is natural to utilize the collected data to improve traffic problems and bring convenient transportation services to people. In other words, the target is to make transportation intelligent from collected spatio-temporal data. One main way to achieve the goal is based on traffic prediction using spatio-temporal data. Thus, the traffic prediction problem has attracted much

attention of both academic and industry. Moreover, with the help of big data and artificial intelligent, there exists a wide spectrum of work on the traffic prediction problem. In this paper, we aim to give a comprehensive survey on the traffic prediction problem, from the collected spatio-temporal data to many intelligent transportation applications.

First of all, it is significant to understand what the traffic prediction problem means. Therefore, we will use some examples to show the concept of traffic prediction:

- *Traffic status prediction*: It is popular to use the navigation system of the electronic map to avoid congested roads when we plan to leave one place for another. The key ability to achieve the target is to predict which roads will be congested in the future time. In other words, we need to predict the traffic status for each road. However, it is typical to measure traffic status with average traffic speed or travel time. The slower the traffic speed or the more the travel time, the worse the traffic status. There-

✉ Guoliang Li
liguoliang@tsinghua.edu.cn

Haitao Yuan
yht16@mails.tsinghua.edu.cn

¹ Department of Computer Science, Tsinghua University, Beijing, China

¹ <https://www.didiglobal.com/>.

² <https://www.uber.com/>.

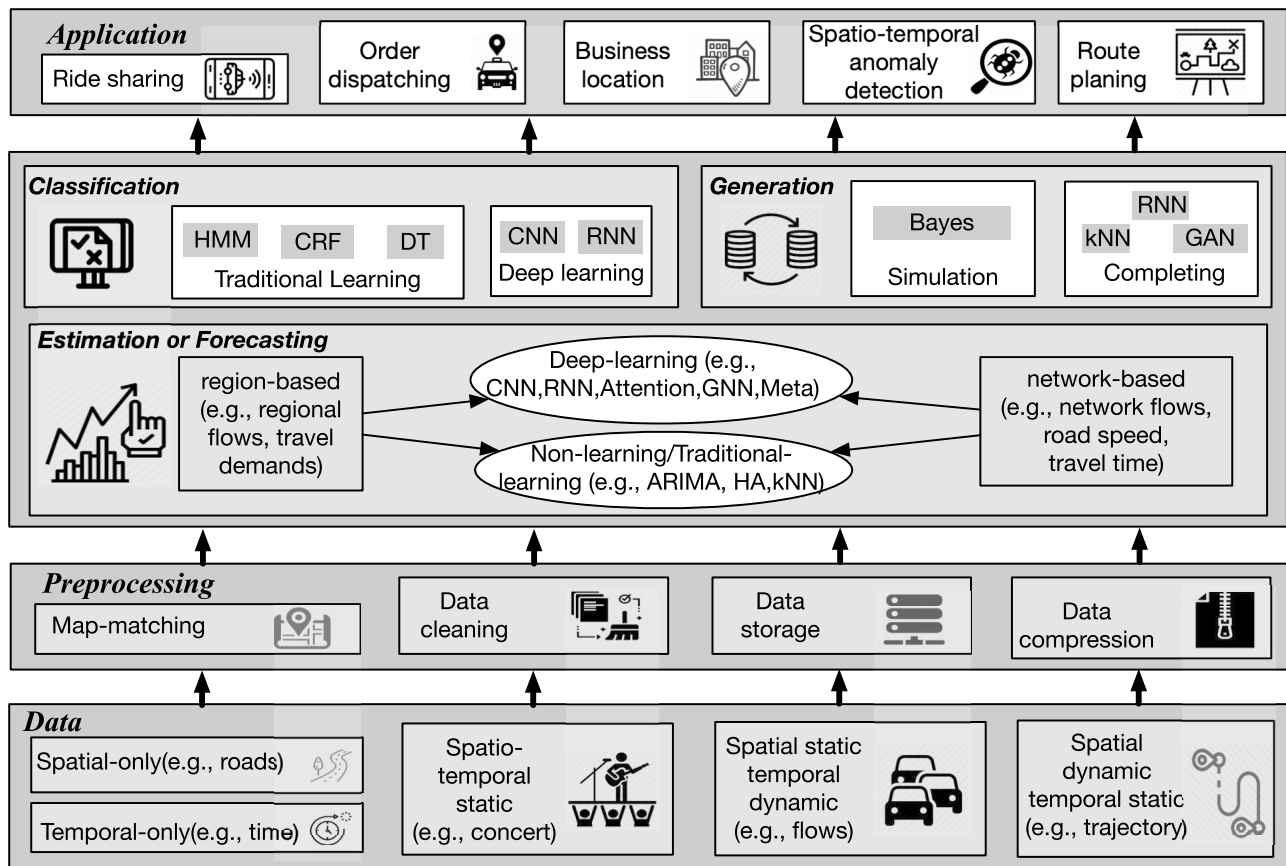


Fig. 1 The overview of traffic prediction

fore, the traffic status prediction can be regarded as the traffic speed or travel time prediction, which are regression problems. Moreover, we can measure the traffic status with different types (e.g., smooth, light congestion and heavy congestion) by splitting the traffic speed into different continuous intervals, where predicting the traffic status becomes a classification problem.

- **Traffic flow prediction:** Recently, there exist some stop events caused by excessive traffic. The main reason is that the government cannot monitor and guide the flow of people in time. Hence, it is significant to predict traffic flows in future time. Moreover, traffic flow can be divided into two types: network-based and region-based. The first type infers the number of vehicles collected by loop detector sensors, which are installed on both endpoints of the roads. As for the second type, we split the whole city into different regions and regard the number of crowds leaving one region for another as the region-based traffic flow. Therefore, the region-based traffic flow can be further divided into in-flow and out-flow. For example, if there are 100 people leaving the region A for the region B, both A's out-flow and B's in-flow would increase 100.

- **Travel demand prediction:** Transportation companies provide online taxi service for users. They need to predict people's travel demands in order to better dispatch vehicles for different regions. For example, they should dispatch more vehicles to residential areas during the morning rush hour. In contrast, they should dispatch more vehicles to office zones during the evening rush hour. Generally, predicting travel demands is based on regions, so we also call it region-based travel demand prediction.

In summary, the above three kinds of traffic prediction problems, respectively, correspond to perspectives of the following three groups: crowds, governments and related companies. Hence, how to solve these traffic prediction problems becomes more and more important in the field of transportation. In other words, the traffic prediction is the indispensable way to make transportation intelligent based on spatio-temporal data. Therefore, we survey the traffic prediction problem by looking from spatio-temporal data to intelligent transportation applications in this paper. As shown in Fig. 1, we mainly consider four parts: data, preprocessing, traffic

Table 1 Comparison of different spatio-temporal data

	Spatial	Temporal	Components
<i>SO</i>	Static	None	POI information, road network
<i>TO</i>	None	Static	Holiday, date, timestamp
<i>STS</i>	Static	Static	Event data
<i>SSTD</i>	Static	Dynamic	Flow, demand, travel time, Velocity, meteorological data
<i>SDTD</i>	Dynamic	Dynamic	Trajectory

prediction problems and traffic applications. Next, we will give a brief overview on four parts.

Spatio-temporal data. As mentioned before, the traffic prediction is based on collected spatio-temporal data, such as road network and historical traffic data. Actually, the spatio-temporal data, related to traffic prediction problems, include road network, region-based traffic (e.g., regional flows and travel demands), network-based traffic (e.g., intersection flows, road speed and travel time), trajectory, POI features, event data, meteorological data and temporal data (holiday, date and timestamp). As shown in Table 1, inspired by [1, 2], the above spatio-temporal data can be broadly categorized into the following five categories:

- (1) *SO* (Spatial-Only data). The data only include spatial features and have no temporal features, such as the POI information and the road network in a city.
- (2) *TO* (Temporal-Only data). The data only include temporal features and have no spatial features, such as the date, the timestamp and the holiday.
- (3) *STS* (Spatio-temporal Static data). For this kind of data, there is no change in both the spatial dimension and the temporal dimension, such as event data.
- (4) *SSTD* (Spatial Static Temporal Dynamic data). The data can be regarded as a sequence in the temporal dimension and keep static in the spatial dimension. The components include traffic flows, travel demands, travel time, traffic speed and meteorological data (i.e., weather).
- (5) *SDTD* (Spatial Dynamic Temporal Dynamic data). The data can be regarded as a sequence in both the spatial and temporal dimensions, such as trajectories. We will discuss more details in Sect. 2.

Preprocessing. Before using collected data to solve the traffic prediction problem, we need to preprocess the data, which involves map-matching, data cleaning, data storage and data compression as follows.

- *Map matching:* Map matching is an operator to convert spatial data with latitude/longitude coordinates into

road networks. For example, we can use map matching techniques to convert a taxi's trajectory (a.k.a., GPS sequence) into a road sequence, by which we can further compute traffic flows on the corresponding roads. Hence, it is significant to apply effective map matching methods for collecting traffic data.

- *Data cleaning:* It is inevitable to generate errors when collecting spatio-temporal data. For example, GPS points may be shifted from their real positions. Hence, through the data cleaning technology, we can correct historical GPS points for predicting the future traffic.
- *Data storage:* With the increase in collected spatio-temporal data, it is non-tractable to efficiently manage them. For example, some travel time prediction methods leverage the average travel time of similar historical trajectories, so efficiently finding similar trajectories is significant for these methods. Here, we aim to survey different methods focusing on how to store and retrieve big spatio-temporal data.
- *Data compression:* Big spatio-temporal data would cause heavy overhead for communication, computing and storage. However, some traffic prediction problems do not really need all data. For example, when computing the region-based traffic flows, we only need to record the number of trajectories coming from one region to another, so it is insignificant to record the whole trajectory information. To address this issue, one method is to compress spatio-temporal data. Here, we aim to survey different methods focusing on how to effectively and efficiently compress spatio-temporal data.

In summary, the quality of preprocessing collected data has great influence on the effectiveness of solving traffic prediction problems. Hence, we will elaborate on the detail of existing work in Sect. 3.

Traffic prediction problems. Generally, there are three kinds of traffic prediction problems—traffic classification, traffic generation and traffic forecasting. Absolutely, the three kinds of problems correspond to three kinds of prediction tasks, which can be summarized as follows.

- *Traffic classification:* The traffic classification problem focuses on how to design effective methods to classify given traffic data. For example, given a taxi's ongoing trajectory, we can use some classification methods to judge whether the trajectory is normal or not and thus can remind the driver to correct the route in time. This is a typical binary classification task. Also, there exist some multiple classification problems. For instance, different modes of transportation (e.g., walking, bus, subway and taxi) should generate different kinds of trajectories. Therefore, given different kinds of trajectories, it is also significant to divide them into different kinds

of modes. To solve the classification problem, existing studies mainly focus on machine learning methods. More specifically, these machine learning methods can be split into two kinds: The first is called traditional learning methods, such as HMM (hidden Markov model [3]), CRF (conditional random field [4]) and DT (decision tree [5]), while the second is called deep learning methods, such as CNN (convolutional neural network [6]) and RNN (recurrent neural network [7]).

- *Traffic Generation*: Obviously, the traffic generation problem means generating some traffic data. The reason of studying this problem is threefold. Firstly, with the development of deep learning techniques, more and more deep learning models are designed to solve traffic prediction problems, and these models require large scale of training data to improve their accuracy. However, it is not easy to collect real-world traffic data for ordinary people, so generating data is an effective way to address this issue. Secondly, some applications (e.g., ride-hailing and taxi dispatching) need to evaluate some approaches on a transportation environment. However, it is unrealistic to use real-world environment due to the lack of all kinds of real-world traffic data. Hence, it is useful to simulate the environment by generating some kinds of traffic data. Thirdly, we need to consider privacy protection when using collected real-world data to train traffic prediction models. Therefore, how to avoid disclosing users' privacy without reducing the effectiveness of trained models is one of the research hot spots. In summary, these reasons make the generation problem split into two parts. One is called simulation, while the other is called completing. For the target of simulation, we try to use collected data to simulate the transportation environment, where we would infer the distribution of traffic data and generate unseen data from other sparse data. Hence, some machine learning methods, such as Bayes [8], are used to generate data or data distributions. As for the target of representation and modeling, we try to model and represent traffic data with hidden codes, from which we can complete unavailable or sensitive data with fake data. More specifically, there are mainly deep learning methods, such as KNN (K-nearest neighbors) [9], GAN (generative-adversarial networks) [10] and RNN.
- *Traffic Forecasting*: The last significant prediction task is to forecast the value of some traffic data, such as traffic speed, traffic flows, travel demands and travel time. Actually, all of these problems belong to two categories, region-based and network-based, according to traffic data's formats. Firstly, in region-based problems, we regard a city as different disjoint regions and compute or estimate related traffic data (e.g., regional flows and travel demands) for each region. For example, the government needs to monitor the crowd flows from one

region to another for avoiding the public security problem caused by the over gathering of crowds. Secondly, in network-based problems, we would consider the constraint of road networks. Specifically, these traffic data (e.g., intersection flows, road speed and travel time) are related to road networks. For example, when we plan to go from one position to another, we would prefer to select the route whose travel time is the least. Here, the travel time should be estimated by designing some effective models.

In summary, traffic prediction problems have a wide coverage, and we will elaborate on the detail of existing work in Sect. 4.

Traffic application. How can we benefit from traffic prediction? The basic answer is to implement rich and varied traffic applications, such as ride-hailing, taxi dispatching, business location, anomaly detection and route planning, based on which the transportation of our city would also be intelligent.

- *Order dispatching*: It is more and more popular to enjoy online taxi services, which are provided by transportation companies, such as Uber, Didi and Lyft. One core problem is to effectively and efficiently assign large scale of taxi orders to drivers. Given large scale of orders, we should design methods to solve the dispatching problem for getting a global optimal solution.
- *Ride sharing*: Ride sharing is becoming a popular mode of transportation with profound effects on the industry. Recent. Given a sharing request, we could estimate the travel time from each candidate car's location to the pickup and then assign the request to the one with the least travel time. However, it is time-consuming to traverse all available candidates. Therefore, when considering larger requests, we need to design more complex methods to make the trade-off between effectiveness and efficiency.
- *Business location*: With the development of smart city, it is more and more popular to leverage find right location to set up a shop or restaurant. Here, one possible solution is based on the crowd flow prediction of regions. Intuitively, the larger the crowd flows are, the better the regions are. In addition, this also can benefit the selection of billboard locations.
- *Spatio-temporal anomaly detection*: Actually, we can convert the anomaly detection problem into a two classification problem and then apply some traffic classification methods to solve the problem.
- *Route Planning*: It is useful to recommend an optimal route for a given departure-destination pair. Similar to taxi dispatching, we can select the route, whose travel time is the least, as the recommendation. Here, we should predict the travel time.

In summary, many applications based on different traffic prediction are used to make our transportation convenient. We will elaborate on the detail of existing work in Sect. 5.

Contribution and Paper Structure. In this paper, we survey a wide spectrum of work on traffic prediction problems as shown in Fig. 1. First, we review different types of traffic data in Sect. 2. Second, we review how to preprocess (e.g., store, compress, clean and map-match) these traffic data in Sect. 3. Third, we divide existing traffic prediction problems into different kinds and then review related methods in 4. Fourth, we review some transportation applications for showing the intelligence of current transportation. Fifth, we provide emerging challenges of traffic prediction in Sect. 6. Finally, we conclude the paper in Sect. 8.

Difference with Existing Surveys. Although there are some surveys [1, 11–19], they only focused on some aspects of traffic prediction, but did not give a complete survey and did not cover most recent works. At first, Wang et al. [18] only survey the management and analytics of trajectories, which is one kind of spatial dynamic temporal dynamic data, so they lack the discussion on other kinds of spatio-temporal data. Similarly, many other surveys just focus on one special kind of traffic prediction problems. For example, Tang et al. [19] focus on the methodology review about the clearance time prediction of road incidents, while the authors in [1, 11, 16] just focus on surveying the traffic flow prediction using machine learning methods. Hence, they cannot give a broad review on the whole domain of traffic prediction. In addition, the authors in [15, 17] survey data mining tasks based on spatio-temporal data, instead of traffic prediction. At last, the authors in [12–14] only give a brief survey on some traffic estimation problems and ignored many current related work.

2 Spatio-Temporal Data

In this section, we first give a figurative example to explain all spatial-temporal data we can leverage for the traffic prediction. Then, we study some existing related work, from which we can deduce the difference of used data for different traffic prediction problems.

2.1 Data Example

As shown in Fig. 2, there is a road network, whose roads with different traffic status are painted in different colors. In particular, we use three kinds of colors (i.e., green, yellow and red) to, respectively, denote three kinds of traffic status (i.e., smooth, light congestion and heavy congestion). In addition, we sample five points, which are marked with A, B, C, D and G, respectively. The difference among these points is that A, B and G are three road interactions, while

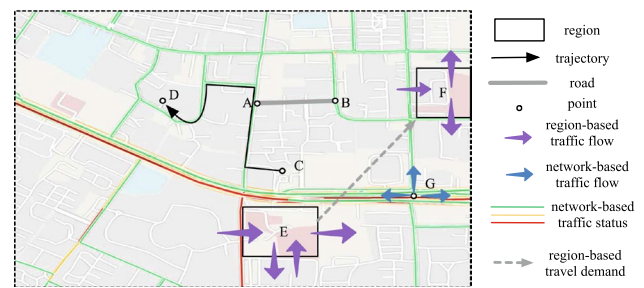


Fig. 2 The road network at a certain moment

C and D are not. Specifically, there is a trajectory stating from C to D. Also, we sample two regions, denoted as E and F, to show region-based traffic prediction. On the one hand, the gray dashed line linking E and F means the region-based travel demand. On the other hand, the purple arrows represent region-based traffic flows. Moreover, each region contains some POI information (e.g., bus stations), which are also related to the traffic prediction. Inspired by [20, 21], we can also incorporate context data for traffic prediction, such as event data (e.g., traffic accident) and meteorological data (e.g., weathers). At last, traffic is changed over time, so we need to consider the temporal data, such as holiday, date and timestamp.

Therefore, as mentioned before, the spatial-temporal data mainly include road network, POIs, region-based traffic, network-based traffic, trajectory, event data and temporal data. In particular, as shown in Table 1, we can further divide them into five kinds: *SO* (Spatial-Only data), *TO* (Temporal-Only data), *STS* (Spatio-Temporal Static data), *SSTD* (Spatial Static Temporal Dynamic data) and *SDTD* (Spatial Dynamic Temporal Dynamic data).

2.2 Reviewing Related Work

At first, Zheng et al. [2, 22] propose the concept of urban computing, which focuses on all computing problems of a city, including traffic prediction problems. Also, they list many related spatio-temporal data, such as geographical data (POIs and road network) and traffic data. In particular, they further split these data into two kinds: point data and network data. For example, POIs belong to point data, while road networks are network data. Later, Zheng [23] only focus on trajectory data mining problems. Hence, they study how to manage and analyze trajectory data. Specifically, when solving the travel time of a given trajectory, the methods can be divided into two groups depending on the availability of the data source: One is called loop-detector-data approach [24–26], and the other is called floating-car-data approach [27, 28]. In other words, they further divide the trajectory data into loop-detector-data and floating-car-data. The loop-detector-data means the data are collected

by loop detectors built under the cross of roads, while the floating-cat-data are collected by sampling from cars' GPS points. At last, when surveying the field of spatio-temporal data mining, Atluri et al. [29] consider four kinds of spatio-temporal data: event data, trajectory data, point reference data and raster data. Here, point reference data correspond to traffic or meteorological data collected at moving ST reference sites (e.g., measuring surface temperature using weather balloons), while raster data correspond to traffic or meteorological data collected at fixed ST grids (e.g., air quality of Earth's surface collected by ground-based sensors). Similarly, Wang et al. [17] follow and extend the work in [29] and classify the spatio-temporal data into five types: event data, trajectory data, point reference data, raster data and videos. However, when reviewing the view data, they just focus on reviewing related works from the perspective of data mining and video data analysis falls into the research areas of computer vision and pattern recognition, and hence, they do not cover the spatio-temporal data type of videos.

In summary, spatio-temporal data contain various types of spatial- and/or temporal-related data, and all of them can be divided into five kinds (*SO*, *TO*, *STS*, *SSTD* and *SSTD*) we have mentioned.

3 Preprocessing

In this section, we, respectively, elaborate map-matching, data cleaning, data storage and data compression techniques for spatio-temporal data.

3.1 Map-matching

The map-matching technique is design to convert spatial data with latitude/longitude coordinates into road networks. Therefore, we only need to focus on these spatial data with the representation of latitude/longitude coordinates, such as trajectories. Existing surveys [23, 55, 56] have surveyed many existing map-matching techniques. For example, Xi et al. [55] split the problem into two types: position matching and curve matching. However, this survey is too old to cover many latest work. Similarly, Zheng et al. [23] and Chao et al. [56] also ignore some existing effective methods.

In this paper, we would bring a broader view on the map-matching problem. As shown in Table 2, we divide existing methods into five types of techniques: Point-Distance, Path-Distance, Probability-Based, Model-Based and Learning-Based. To better compare these methods, we provide three features: *Geometric*, *Topological* and *Global*. *Geometric* means that we should consider the geometric information of spatial data, such as the Euclidean distance. *Topological* means that we should consider the topological

Table 2 Comparison of different map-matching methods

Technique	<i>Geometric</i>	<i>Topological</i>	<i>Global</i>	Examples
Point-Distance	Yes/no	Yes/no	No	[30–34]
Path-Distance	Yes/no	Yes	Yes	[35–40]
Probability-Based	No	Yes	Yes	[41–43]
Model-Based	Yes	Yes	Yes/no	[44–48] [49–52]
Learning-Based	Yes	Yes	Yes/no	[53, 54]

structure constraint of road networks. *Global* means that we should consider the global optimal matching instead of greedy local optimal solutions for matching a sequence of GPS points onto roads.

Firstly, the Point-Distance methods leverage some distance functions to match sampled GPS points on road networks. In particular, some methods [30, 31] just consider matching a GPS point on the nearest road by computing the Euclidean distance, and some work [32–34] focus on a trajectory by sequently matching each point of the trajectory onto a road with some greedy strategies. Differently, Quddus et al. [32] compute the shortest path between sampling points when finding the next matched road. Totally, all Point-Distance methods ignore the *Global* feature.

Secondly, the Path-Distance methods [35–40] are designed to match trajectories onto road networks. Specifically, they compute the similarity between a partial/whole trajectory with its matched road/path, and the similarity is measured by the distance between a trajectory and its matched path. Some works [38, 39] aim to match an entire trajectory with a road network by computing the Euclidean distance. Differently, other work leverage sequence similarity functions to compute the distance. For example, Fréchet distance is the most commonly used distance function [35, 40] since it considers the monotonicity and continuity of the sequence. However, this distance can be dominated by these noisy points when a trajectory includes many noisy points. To address this issue, Zhu et al. [36] leverage the LCSS (longest common subsequence) function to compute the similarity, where they select the matched route, who has the maximum LCSS similarity, as the final result. In addition, Zheng et al. [37] use historical map-matched data to answer new map-matching queries by assuming people tend to travel on the same path when given origin and destination points. In particular, for a given trajectory, they first find similar historical trajectories as candidates and then use a scoring function to decide the optimal route.

Thirdly, to improve the robustness of map-matching, Probability-Based methods [41–43] make explicit provisions for GPS noise and consider multiple possible paths through the road network to find the best one. In particular, Ochieng et al. [41] develop an improved probabilistic

map-matching algorithm, whose main characteristic is taking into account the error sources associated with historical trajectory of the vehicle and topological information on the road network and so on. Differently, pink et al. [43] represent the road network topology with a stochastic finite state machine, where every edge in the digital map is represented by one state for each driving direction, and then, they estimate the distribution with historical data. At last, fuzzy logic is one technique that is an effective way to deal with qualitative terms, linguistic vagueness and human intervention. Quddus et al. [42] develop a map matching algorithm based on fuzzy logic theory, where the inputs are from the global positioning system augmented with data from deduced reckoning sensors to provide continuous navigation.

Fourthly, Model-Based methods leverage some powerful models to solve the map-matching problem, such as partial filter [47, 49], HMM (hidden Markov model) [44–46, 48, 50], CRF (conditional random field) [51] and WGT (weighted graph technique) [52]. PF is a local optimal model. Specifically, PF is to recursively estimate the probability density function (PDF) of the road network section around the observation as time advances. In other words, once getting a new observation, the PDF for the road network section around the new observations is calculated and the area with the highest probability is determined as the matched region. Differently, HMM, CRF and WGT are three kinds of global optimal models. At first, HMM is the most popular used model, which simulates the road network topology and meanwhile considers the reasonability of a path. They regard the sampled trajectory as the observation and the vehicle actual location on the road, which is unknown, are the hidden states. The major difference between various HMM-based algorithms is their definition of emission probability and transition probability. For example, Some works [45] prefer a candidate pair whose distance is similar to the distance between the observation pairs, while others consider velocity changes [48] and turn restriction [50]. To avoid the selection bias problem, Hunter et al. [51] leverage the model CRF. However, both HMM and CRF have no recovery strategies for the match deviation. Since once a path is confirmed, it will be contained by all future candidate paths. To address this issue, the WGT mode, aiming to build a weighted candidate graph for inferring the matched path, is used [52]. In the candidate graph, the edge weight is computed by some score function, so it can be adapted for the matched deviation.

Finally, with the help of historical matched data, there are many work focusing on leverage learning methods to solve the problem. In particular, Sharath et al. [53] learn a score function to evaluate candidate grids around the observed location at each timestamp. Considering the powerful fitting ability of neural networks, Zhao [54] learn a sequence-to-sequence neural network to directly convert a sequence of

locations into a sequence of roads. Notably, the former work is a local optimal method, while the latter is not.

3.2 Data Cleaning

The target of data cleaning is to solve some data problems, which can result in the inaccuracy and inefficiency of traffic prediction. Actually, data problems are composed of data missing, data outlier and data imbalance, so we will review existing work following the three problems.

Data missing: Spatio-temporal data often suffer from missing values due to some complex reasons, such as hardware failures, software bugs and human errors. The direct solution is to fill missing values. For example, Lee et al. [57] design a factorial hidden Markov model to recover missing values, while Yi [58] combine many empirical statistic models (e.g., inverse distance weighting and simple exponential smoothing,) with user-based and item-based collaborative filtering to collectively fill missing value for geo-sensory time series data. However, these methods cannot better capture both spatial and temporal features among readings and unavoidably ignore the global correlations of data. To address this issue, some researchers [59, 60] treat raw data as a matrix and propose various matrix completion/recovery methods to estimate the missing values by capturing their inherent low-rank structure.

Data outlier: Collecting outlier data is another common problem caused by some complex reasons. The process of solving this problem includes two steps: identifying outlier and repairing data. On the one hand, many work are proposed to detect spatial and temporal outliers. Some researchers regard data, whose values are different from their spatial or temporal neighborhoods, as spatial outliers, and then apply different methods to construct local neighborhoods and assign anomaly scores. For example, Knorr and Lu [61, 62] use spatial distance measures to compute anomaly scores for spatial objects; while Shekhar and Kou [63, 64] use graphical distance measures for spatial objects. To extend spatial outlier detection to spatio-temporal data, some researchers [65] leverage some algorithms, such as DBSCAN, to cluster normal data and then report the data with no conformed clusters as outliers. On the other hand, how to repair spatio-temporal data is also discovered by some researchers. For example, Mauder et al. [66] define the dissimilarity between the raw data and its repaired state and then propose some rules of spatial or temporal distortion to minimize the dissimilarity. However, their method only considers local minimum without taking the whole repairing space into account. To address this issue, Zhou et al. [67] propose a novel robust spatio-temporal tensor recovery (STTR) method to deal with both missing data and outliers. In particular, they organize the data as a multi-way array

(i.e., tensor) and incorporate domain knowledge about the structure of the underlying data for repairing anomaly data.

Data imbalance: Data imbalance actually means the imbalance of data distribution or data label. On the one hand, some roads with heavy traffic would collect dense traffic data, while others only correspond to sparse data, and this phenomenon is called data distribution imbalance. On the other hand, there are many vehicle trajectories and few pedestrian tracks when training a classification model, and this phenomenon is called data label imbalance. To handle the data distribution imbalance, Zheng et al. [68] design a semi-supervised learning method, which solve the problem of the sparse training data, which is caused by the lack of air monitoring stations. Also, some researchers focus on solving the data label imbalance. Beckmann et al. [69] first leverage the KNN-based undersampling methods to solve the problem. In addition, Wang et al. [70] further design a K-labelsets ensemble method based on mutual information and joint entropy; while Gong et al. [71] present a ensemble method using random undersampling and ROSE sampling to solve the imbalance classification problem.

3.3 Data Storage

With the increase in spatio-temporal data volumes, how to store these data becomes increasingly challenging. One main solution is to leverage some distributed systems to store data. Hence, based on this characteristic, existing work can be divided into two kinds: based on single machine and based on distributed system.

Single machine: The goal of storing spatio-temporal data is to make it easy to query. Specifically, to achieve this goal, many researchers study how to build index for supporting efficient queries. At first, R-tree [72] is designed to index spatial objects with multi-dimensional information such as geographical coordinates, rectangles or polygons. There are two kinds of approaches to extend R-tree to index spatio-temporal data. The first method regards the time as the third dimension and then build a 3D-Rtree, such as STR-tree and TB-tree [73]. The drawback of this method is that the overlap among different objects still keeps on increasing as time goes by, which would result in the inefficiency of querying data. The second method first splits a time period into multiple time intervals and then builds R-tree index for spatio-temporal data in each time interval. In particular, if some parts of an index are not changed over time, they would be shared by different time intervals. In particular, the representative index structure is multiple version R-tree, such as Rt-tree [74], HR-Tree [75] and H+R-tree [76]. Another popular structure for indexing spatial data is grid index [77]. Intuitively, they split the spatial space into disjoint grids, and thus, different spatial objects would belong to different grids. Also, Wang et al. [78] extend this structure to

support spatio-temporal data (e.g., trajectories). In addition, some queries focus on road networks. For example, Zhong et al. [79] propose the G-tree structure to manage road vertices and support for efficiently finding the shortest path between any two road vertices on a road network.

Distributed system: Recently, how to achieve parallel computation with multi-machines has attracted many researchers' attention. One popular parallel framework is called Map-Reduce, based on which the distributed system Hadoop [80] is created. Later, two distributed systems Spatial-Hadoop [81] and Hadoop GIS [82] are designed for spatial data analytics, where the two systems are implemented based on Hadoop. To support spatio-temporal data analytics, Tan et al. [83] further design a Hadoop-based storage system, which is called Clost. With the popularity of in-memory computing, a new distributed system Spark [80] is proposed. Spark has its architectural foundation in the Resilient Distributed Dataset (RDD), a read-only multi-set of data items distributed over a cluster of machines. The latency of Spark-based applications may be reduced by several orders of magnitude compared to Hadoop MapReduce implementation. Naturally, some researchers extend Spark to support managing spatio-temporal data. For example, GeoSpark [84] and Simbda [85] are two useful systems for processing spatial data. Differently, GeoSpark does not support Spark SQL [86] or the DataFrame API, while Simbda can support. In addition, there are some works [87–89] focusing on distributed trajectory analytics with Spark, such as similarity search and join. Differently, Yuan et al. [89] consider the structure of road networks when processing trajectory analytics.

3.4 Data Compression

Sometimes, due to the heavy cost of communication, computing and data storage, it is unnecessary to record all spatio-temporal data in a fine-grained manner, especially when collecting the trajectory data for a moving object. To save cost with reducing a litter precision, many researchers have studies how to compress trajectories data. Depending on whether the fully trajectory is generated before the process of compression, these works can be divided into two types: *Offline* and *Online*.

Firstly, *Offline* methods can be further divided into two categories: *simplification-based* and *road network-based*. *Simplification-based* methods aim to reduce some unnecessary points from the raw trajectory data. For instance, Douglas–Peucker algorithm [90] iteratively uses an approximate line to replace the raw trajectory until the error is beyond a given threshold. In addition, the authors in [91, 92] directly remove extra points from trajectory when the sampling rate is high. However, this way can reduce the resolution of data analytics. To address this issue, Zhang et al. [92] also define

an error ratio to bound the loss of simplification, which is similar to Douglas–Peucker algorithm. *Road network-based* methods enhance the quality of compression using the road network. The authors in [93] match each trajectory onto roads and then represent it with a sequence of roads. Later, they use *Huffman coding* to represent each road, and thus, a trajectory would be represented as a concatenation of the codewords and is significantly more effective than raw data. Regarding the sequence of roads, the authors in [94] apply string compression methods to solve the problem.

Secondly, *online* methods aim to compress trajectory data in a timely fashion. There are two types of algorithms: *window-based* and *moving attribute-based*. In particular, *window-based* algorithms [95, 96] maintain a growing sliding window for fitting spatial points with a line segment and continue to grow the sliding window until the approximation error exceeds some error bound. Differently, *moving attribute-based* algorithms consider the attributes of moving objects, such as speed and directions, as main factors for online compressing trajectories. For example, Potamias [97] uses last two locations and a given threshold to build a safe area. If a new spatial point is located in the safe area, they consider the point as redundant, and thus, discard it; otherwise include it in the final trajectory.

4 Traffic Prediction

Traffic prediction problems include three types: traffic classification, traffic generation and traffic forecasting. In this section, we will elaborate existing work on these problems.

4.1 Traffic Classification

Traffic classification means leveraging different methods to classify given spatio-temporal data, such as GPS points and trajectories. In particular, according to the difference of used techniques, related work can be split into two types: traditional learning methods and deep learning methods.

Traditional learning. One important traffic classification problem is to detect transportation modes based on given spatio-temporal data. In particular, given collected transportation information of a moving object, the corresponding task is to classify the motion of the object. For example, Krumm et al. [98] leverage the hidden Markov model, which takes the sequence of wireless signals as the inputs for a device, to determine whether the device is moving or not. Also, Timothy et al. [99] use the hidden Markov model to categorize a user's mobility into three types: stationary, walking and driving. In most cases, one single trip may contain some different transportation modes, so many researchers first split each trip into different segments and then leverage different methods to classify each segment into different

modes. For instance, Zhu et al. [100] aim to monitor the status of a taxi. They define the status with three states: Occupied, Nonoccupied, and Parked. Specifically, given a trajectory of a taxi, they first find Parked points and then split the trajectory by these Parked points. Later, they extract features (e.g., road networks and points of interest (POIs)) and locally learn a probability classifier to classify each segment into either Occupied or Nonoccupied. Globally, they apply a hidden semi-Markov model to mining travel patterns. Similarly, Zheng et al. [101, 102] split a trajectory into continuous segments and then design decision tree classifier to classify each segment into four kinds: Driving, Biking, Bus and Walking. Here, the extracted features for each segment include the heading change rate, stop rate, and velocity change rate. Considering that GPS points are sampled from cars passing through road networks, Liao et al. [103] and Patterson et al. [104] first divide trajectories into 10-m segments and then leverage CRF (conditional random field) model to map-match segments onto road networks. Hence, they use matched road information to classify raw trajectory into a sequence of activities (such as Walk, Driving and Sleep) and identify the corresponding user's significant places (e.g., home, work and bus stops), simultaneously. Differently, Yin [105] designs a hierarchical DBN (dynamic Bayes network) model to detect the sequence of activities based on a user's wireless signals, where the high-layer class is inferred based on lower-layer inferred results. At last, Stenneth et al. [106] propose a transportation mode detection framework by integrating collected GPS information and knowledge of the underlying transportation network, where the transportation network information include real-time bus locations, spatial rail and spatial bus stop information. Based on this framework, they can apply five models, a.k.a., Bayesian net, decision tree, random forest, Naïve Bayesian and multilayer perceptron, to distinguish between motorized transportation modes such as bus, car and aboveground train with such high accuracy.

Deep learning. With the development of deep neural networks, many researchers try to apply different deep learning methods to solve the traffic classification problem. In particular, these methods, respectively, belong to three types: CNN-based and RNN-based.

CNN-based. The convolutional neural network (CNN) technique plays an important role in improving the classification accuracy of images [107]. Therefore, many researchers in the domain of intelligent transportation try to leverage CNN techniques to solve transportation classification problems related to images. For example, Nolte et al. [108] focus on the condition of the road surface and train two different convolutional neural network models to classify the photo taking on the road surface, which helps enabling an early parameterization of vehicle control algorithms. Similarly, Ramanna et al. [109] leverage CNN techniques to

classify photographs taken from road cameras by weather conditions, where photographs would be labeled as dry, wet, snow and so on. Pamula et al. [110] try to detect the traffic condition based on video surveillance data. Here, they also leverage CNN to classify the traffic condition based on observed video contents. *RNN-based*. The recurrent neural network (RNN) technique is designed to model sequential data. Therefore, there are some works studying how to apply this technique to solve the classification problem in the domain of transportation. At first, Liu et al. [111] apply the RNN model to solve the transportation mode classification problem. In particular, they design an end-to-end classification framework based on the bidirectional LSTM (long short-term memory), which is one kind of RNN architecture. Also, Qin and Nawaz [112, 113] apply the LSTM model to recognize or learn transportation modes. Differently, before using the LSTM model to capture the temporal dependencies characteristics on the feature vectors, they first use a CNN model to learn appropriate and robust feature representations for transportation modes recognition. To accelerate the learning speed and enhance the accuracy of transportation mode detection, Wang et al. [114] utilize the residual architecture [115] beyond the LSTM model. Finally, Liu et al. [116] consider both spatial information and temporal information for trajectory classification. They apply another RNN architecture GRU to model the spatio-temporal correlations and irregular temporal intervals prevalently present in spatio-temporal trajectories.

4.2 Traffic Generation

Traffic generation is an important way to simulate transportation environments and provide sufficient data for other traffic prediction problems. Hence, all related works belong to two types: simulation and completing. Simulation aims to generate some data to simulate actual scenarios based on historical observations, while completing means generating data to represent unavailable data for other prediction problems.

Simulation. Most researchers study the platform construction of traffic environments. In particular, they first use Bayes technique to compute related data distribution based on historical traffic data and then use the distribution to simulate different traffic conditions. For example, Brinkhoff et al. [117] produce a platform to generate moving objects, where they combine a real network with user-defined properties of the target dataset. In [118], a simulator is presented to help to prepare and to perform the simulation of traffic scenario, which includes network generation, demand generation and traffic generation. Similarly, Lon et al. [119] design a specialized platform to test algorithms for pickup-and-delivery problems. Also, Adnan et al. [120] design a simulator to model millions of agents over a large

range of mobility decisions. At last, The simulator proposed in [121] is designed specifically for ridesharing by including components and routines common to ridesharing algorithms.

Completing. On the one hand, some people study how to generate an individual data for solving other prediction problems. For example, Wang et al. [122] generate routes to estimate the travel time from an origin point to a destination point. They leverage the kNN technique to find the nearest historical route, whose origin and destination points are similar to given origin and destination points, to compute the travel time. However, the kNN technique cannot work when historical data are too sparse. To address this issue, Song et al. [172] leverage GAN (generative adversarial networks) to generate human mobility routes. They design two representative discriminator and generator networks, where the discriminator network contains four layers of convolutional neural networks for capturing essential location features. On the other hand, some people focus on the modeling of spatio-temporal data, by which they can generate fake data to replace actual data for avoiding privacy disclosure. For example, Wu et al. [173] model the trajectory data with RNN and hence encode a trajectory into a hidden code. In particular, they regard each trajectory as a road sequence and then make full advantage of the strength of RNN to capture variable length of the sequence. Meanwhile, they consider the constraints of topological structure on road networks when modeling trajectories.

4.3 Traffic Forecasting

The forecasting problems prefer to predict certain future traffic states. As shown in Table 3, we survey six types of problems: OD-Travel-Time, Path-Travel-Time, Travel-Demand, Regional-Flow, Network-Flow and Traffic-Speed. In particular, existing related work can be roughly divided into two categories: *non-learning* and *learning* methods. More specifically, *learning* methods can be further divided into *traditional-learning* and *deep-learning* methods. In details, these methods contain different techniques. For example, *non-learning* methods include kNN and HA (historical average), and *traditional-learning* methods include regression, DT (decision tree) and HMM (hidden Markov model). In addition, five features (i.e., road network, environmental data, spatial property, temporal property and nonlinearity) are considered when reviewing these techniques. Firstly, the structure of road network is a significant constraint when handling traffic prediction on roads or intersections. Secondly, environment data, such as weather, play an important role in traffic prediction. Thirdly, spatial properties (e.g., POIs, roads and maps) also influence the traffic. For example, the traffic in business district is totally different from the traffic in residential district. Fourthly, temporal properties (e.g., holiday information,

Table 3 Comparison of different traffic forecasting methods

Problem types	Techniques	Examples	Consider road network	Consider environmental data	Consider spatial property	Consider temporal property	Handle nonlinearity
OD-	kNN	[122]	×	×	×	×	×
Travel-	MLP	[123]	×	×	×	×	✓
Time	ResNet,LSTM,CNN	[124, 125]	✓	✓	✓	✓	✓
Path-	kNN,TD,Regression	[126–129]	✓	×	×	×	×
Travel-	DT,HMM	[130, 131]	✓	×	×	×	✓
Time	CNN,LSTM,W-D	[132–134]	✓	✓	✓	✓	✓
	Generative	[28, 135–137]	✓	×	×	×	✓
	HA,ARIMA,ensemble	[138, 139]	×	×	×	×	×
Travel-	MLP	[140]	×	✓	×	✓	×
Demand	CNN+RNN	[21, 141–145]	×	✓	✓	✓	✓
	GCN,GAT	[146–149]	✓	✓	✓	✓	✓
Regional-	HA,ARIMA,ensemble	[138, 150]	×	×	×	×	×
Flow	CNN	[151]	×	✓	×	×	✓
	CNN+LSTM	[152, 153]	×	✓	×	×	✓
	HA,ARIMA,ensemble	[138, 154–157]	×	×	×	×	×
Network-	Autoencoder	[158]	×	×	×	×	✓
Flow	GCN,Attention	[159–161]	✓	×	×	×	✓
	GCN+RNN	[162, 163]	✓	×	×	×	✓
	Meta-learning	[164]	✓	×	✓	×	✓
Traffic-	HA,ARIMA	[138]	×	×	×	×	×
Speed	CNN,LSTM,FNN	[165–170]	✓	×	×	×	✓
	LSTM+GCN	[171]	✓	✓	✓	✓	✓

events) may be useful for the effectiveness of traffic forecasting. For example, the pattern of traffic on weekends is different from that on weekdays. Fifthly, there exists complex nonlinearity relationship between inputs and outputs when estimating future traffic, so whether handling nonlinearity is one way to measure the effectiveness of different forecasting methods. In summary, we survey existing work based on the above five features as follows.

OD Travel Time. The OD-Travel-Time problem aims to estimate the travel time for a given OD input, which consists of an origin point, a destination point and a departure time. At first, the authors in [122] leverage the kNN technique to select historical trajectories, whose origin and destination points are similar to given OD input and then compute the average value of selected trajectories as the estimated result. Later, some people utilize the deep neural network to solve the problem. MLP (multilayer perceptron), also known as multilayer fully connected neural network, is used to estimate the OD travel time in [123]. In particular, the authors first use MLP to estimate the travel distance based on given origin and destination points and then use MLP to estimate the travel time based on the estimated distance and given departure time. However, these methods ignore some features (e.g., the structure of road network),

so other deep neural networks are applied to address this issue. For example, Li et al. [124] use the residual neural network (ResNet) to encode each given OD input, as well as the features about road network, spatial properties, temporal properties and so on. Considering the usefulness of historical trajectories, Yuan et al. [125] utilize LSTM and CNN techniques to design an auxiliary model to encode historical trajectories, by which the estimated travel time would be accurately affiliated to a trajectory. Similarly, they consider the features about the environmental data, the temporal and spatial properties, as well as road networks.

Path Travel Time. The Path-Travel-Time problem is defined to estimate the travel time for a given path/route on road networks. Hence, all existing works consider road networks. At first, similar to OD travel time estimation, Rahmani et al. [126] leverage kNN methods to select nearest neighbors of historical sub-trajectories to compute the travel time. Considering the ineffectiveness of using historical data due to its sparseness, Wang et al. [127] model different drivers' travel times on different road segments in different time slots with a three-dimensional tensor and then fill in the tensor's missing values through a context-aware tensor decomposition (TD) approach. However, this method cannot capture the dynamic of travel patterns. Other people regard

the problem as a linear regression problem [128, 129], which corresponds to learning-based methods and takes as input the given path/route. Differently, the authors in [129] consider the temporal dynamic, which would learn different weights for different time slots. However, both of them learn the linear weight of corresponding regression models, so they cannot handle the nonlinearity. To address this issue, other machine learning methods, such as DT (decision tree [130]) and HMM (hidden Markov model [131]), are applied to solve the problem. In particular, they partition the whole path into a sequence of links and then estimate each link's travel time. The authors in [130] independently estimate each link via some boosting techniques (such as AdaBoost and gradient boosting tree), while the authors in [131] model the whole sequence via the HMM technique. However, these traditional methods ignore some useful features, such as spatial and temporal properties. There are thereby some works focusing on leveraging deep learning techniques (e.g., CNN and LSTM) to solve the problem. For example, the authors in [132, 133] first regard the given route as a sequence of segments. Then, they leverage CNN model to encode each segment for capturing local spatio-temporal correlations, based on which they further leverage RNN model to encode the whole route. Also, they encode external data (e.g., environmental data, spatial and temporal properties) for better estimation. In addition, there is a related work [134] utilizing the Wide-Deep (W-D) model to solve the problem. They divide inputs into different parts, which are, respectively, encoded by different wide (e.g., affine transformation) and deep (e.g., MLP and LSTM) models. At last, some researchers would prefer the distribution of travel time rather than the value. In particular, Hunter et al. [28] model the route by a generative distribution model and then apply the EM (expectation maximization) to learn the model's parameters. Similarly, Asghari et al. [135] learn the travel time probability distributions from historical data for each and every edge/link on road networks and then jointly compute the distribution for the whole path/route. Differently, the authors in [136] avoid blasting trajectories into small fragments and instead assign distributions to paths rather than simply to the edges/links. Also, the authors in [137] apply deep learning methods to generate probability parameters for corresponding generative models.

Travel Demand. The Travel-Demand problem aims to predict the future transportation requests for each region of a city. At first, the ensemble of some basic models is proposed in [139] combining five base learners (i.e., Time-Varying Poisson Process, Fading-Factor TVPP, ARIMA, L1-regularized Vector AutoRegressive process with exogenous variables and Drift-Aware VAR process) to improve the effectiveness. Later, with the help of deep learning, Wang et al. [140] apply the MLP model and the residual network architecture to forecast both travel supply and travel demand. In addition,

other complicated deep learning techniques (e.g., CNN and RNN) are employed to solve the problem. Specifically, many researchers [21, 141, 143–145] regard the historical traffic demands as a sequence. Then, they leverage CNN models to encode the data at each time step and further leverage RNN (e.g., LSTM and GRU) models to encode the whole sequence for capturing sequential features. The difference among these methods mainly locates in the processing of other useful informations. For example, Yao et al. [21] further encode each region by taking the semantic similarity among regions into account, while [141, 143, 145] further encode contextualized features, such as spatial and temporal properties. In addition, Kuang et al. [142] regard historical traffic demands as a 3D tensor and then apply 3D-CNN model to encode the data, and they also apply multi-task learning technique to enhance the performance. However, the above deep learning methods cannot capture some graph features, such as road networks. To address this issue, some people [146–149] apply graph neural networks (e.g., GCN and GAT) to capture graph features. For instance, Geng et al. [146] build three graphs, respectively, considering neighborhood, function similarity and connectively among different regions, to capture complex spatial dependency. Also, they further apply RNN model to capture temporal dependency.

Regional Flow. The Regional-Flow problem is defined to forecast future traffic flows among regions. At first, the authors in [150] utilize the ensemble of some base learning methods (e.g., AdaBoost and random forest) to predict traffic flows. Later, Zhang et al. [151] first leverage deep learning methods to solve the flow prediction problem. In particular, they split the whole city into disjoint regions and then define inflow and outflow for each region. Regarding historical traffic flows as pictures, where each region corresponds to a pixel, they apply the CNN model to encode them. In addition, they consider some environmental data as external features in their whole model. However, they ignore the sequential characteristics among historical data. Therefore, other researchers regard the traffic data at each timestamp as a picture and regard all timestamps as a sequence of pictures. Specifically, the authors in [152, 153] first apply the CNN model to encode the data in each timestamp and then apply the LSTM model to encode the whole sequence. In addition, considering that the influence of each historical traffic flow has different influences on the future traffic flow, Yao et al. [153] leverage the MLP model to encode the future environmental data and then compute the attention value between the future encoded results with each historical encoded traffic data. Hence, they consider each attention value as the corresponding weight for each historical timestamp, where the weight can represent the influence.

Network Flow. The Network-Flow problem focuses on the flow passing through each intersection on road networks.

Except for general time series forecasting methods (i.e., HA and ARIMA), there are other traditional learning methods. For example, Jin et al. [154] leverage PCA (principal component analysis) and SVR (support vector regression) techniques to predict network traffic flows. They use PCA to reduce the dimension of traffic data by outputting eigenflow data. After that, they apply SVR to predict the eigenflow data, based on which they reconstruct the flow data. Also, Tang et al. [155] leverage the SVR method to solve the problem, but they enhance the method with some denoising algorithms. They further combine one kind of denoising algorithm (ensemble empirical mode decomposition) and the fuzzy C-means neural network (FCMNN) to improve prediction accuracy [156]. To predict for multivariate traffic flows, Yan et al. [157] adopt a weighted Frobenius norm to estimate similarity between multivariate time series, where the weights are determined by the PCA method. Recently, most people rethink the traffic flow prediction based on deep architecture models. At first, Lv et al. [158] use a stacked autoencoder model to learn generic traffic flow features, and the model is trained in a greedy layer-wise fashion. Later, taking into the structure of road networks, the authors in [159–161] leverage GCN models to predict the network flows. In particular, Fang [159] build the spatio-temporal block to encode historical traffic data, where the block contains multi-resolution temporal module and a global correlated spatial module. Wang et al. [160] propose a two stream network, where the first stream corresponds to a novel graph-based spatio-temporal convolutional layer, aiming to extract features from a graph representation of traffic flow, while the second stream predicts the dynamic graph structures, and the predicted structures are fed into the first stream. Guo et al. [161] propose two parts of modules to encode historical data: The first part leverages the attention mechanism to capture the dynamic spatio-temporal correlations in traffic data, while the second part uses the GCN technique to capture the spatial patterns and common standard convolutions to describe the temporal features. Also, some people [162, 163] consider the sequential features among historical network flows, so they further append RNN models to encode historical data. Specifically, Li et al. [162] model the traffic flow as a diffusion process on a directed graph and introduce diffusion convolutional recurrent neural network (DCRNN), a deep learning framework for traffic forecasting that incorporates both spatial and temporal dependencies in the traffic flow. Differently, Wang et al. [160] first leverage spatial GNN to encode historical data and then leverage GRU model to encode the whole sequence. They finally use the transformer model to further encode the output of the GRU model. At last, the meta-learning method is also applied for capturing the dynamic dependency among traffic flow data [164]. The advantage is that they consider the spatial properties of road networks.

Traffic Speed. The Traffic-Speed problem aims to forecast the speed of cars on roads. Similar to other traffic forecasting problems, instead of using general time series prediction methods (i.e., HA and ARIMA), people recently apply many deep learning methods. Firstly, some people only consider apply different deep models to encode historical traffic data. For example, Ma et al. [166] convert the spatio-temporal traffic data into images describing the time and space relations of traffic flow via a two-dimensional time-space matrix, which is encoded by the CNN model. Cui et al. [168] propose a deep stacked bidirectional and unidirectional LSTM neural network architecture, which considers both forward and backward dependencies in time series data, to predict network-wide traffic speed. In addition, a bidirectional LSTM layer is exploited to capture spatial features and bidirectional temporal dependencies from historical data. The authors in [165, 167] take advantage of both RNN and CNN models by a rational integration of them. In particular, they first use the CNN model to capture topology aware features, and then, the periodicity and context factors are also considered to further improve accuracy by applying the LSTM model. To forecast the traffic speed for multi-step ahead, Tang et al. [169] propose an evolving fuzzy neural network with two proposed learning processes, where the first is to cluster inputs and the second is to optimize parameters in the Takagi–Sugeno-type fuzzy rules. Also, similar to [170], they consider the influence of periodic component in the raw speed data. However, the above methods ignore many contextualized features, such as spatial and temporal properties. Therefore, Liao [171] take into many implicit but essential factors for predicting traffic speed, where they integrates these data as follows. Firstly, they consider offline geographical and social attributes, such as the geographical structure of roads or public social events. They apply the GCN model to encode the information. Secondly, they consider online crowd queries, which are regarded as a sequence and encoded by the LSTM model.

5 Traffic Application

Making it possible to achieve intelligent transportation, many applications should be developed based on traffic prediction. In this paper, we survey five broadly used applications, which are, respectively, called ride sharing, order dispatching, business location, anomaly detection and route planning. In addition, these applications heavily rely on the performance of traffic prediction techniques. For example, before dispatching taxi orders, deciding ride sharing strategies or planning routes for users, we should estimate the travel time or traffic speed on road networks. In other words, the more accurate the predicted future traffic states,

the better these traffic application services. Next, we will elaborate them.

5.1 Ride Sharing

More and more people are pleasant to share their ride with others due to the full use of resources and the environmental friendliness. The goal of ride sharing is to maximize the profit or the number of customers being served, which is greatly influenced by traffic states, such as traffic speed and travel time. Hence, accurately forecasting these traffic states would improve the effectiveness of ride sharing algorithms. Similar to the survey in [121], we also review an exact offline method and some online methods.

The actual offline method is called Branch-and-Bound (BB), which is a general method to solve mixed-integer linear programs (ILP). As the ride sharing problem can be formalized as an optimization problem about mixed-integer linear programs, BB can be extended to solve the ride sharing problem [174, 175]. In particular, BB would build a search tree to explore solutions. At first, they construct the tree's root node by solving the relax problem associated with ILP. Later, they iteratively search and construct other nodes for getting optimal solutions as follows: (1) Branch: Create two child nodes for every node that represents a non-integer solution. Each child takes the same relaxed problem as its parent. And both child nodes represent two new relaxed problems, each with one less binary variable. (2) Bound: Solve each new relaxed problem to obtain new solutions.

The online methods include two kinds: search-based and join-based. Firstly, search-based methods would search the optimal matched vehicle for each order with the way of one-by-one. Specifically, Jung et al. [176] select nearest vehicles to assign orders, where they measure them based on distance. However, small distance cannot correspond to optimal matches because inserting customers into some vehicles' schedule would influence vehicles' current customers' routes. Hence, many people [177–179] first try to insert a customer's route into candidate vehicles and then select the vehicle with the least cost to actually insert the customer's route. Considering that the time complexity of trying all candidates is too large, Huang et al. [180] design the kinetic tree (KT) to improve the efficiency. They only remember the valid schedules for a vehicle by pruning invalid ones from the kinetic tree. To improve the quality, Cheng et al. [181] consider a *replace* procedure when matching orders and vehicles. Secondly, join-based methods would batch orders into a set and then assign orders all at once. More specifically, the join-based methods consist of two kinds of frameworks: the initialize-improve framework and the group-assign framework. In the initialize-improve framework, people usually use a heuristic method to get a set of initial assignments and then try to use additional procedures

to improve the assignments. For example, people have applied simulated annealing (SA), a single-solution meta-heuristic for general optimization problems, to the ride sharing problem [176]. In particular, they randomly initialize the assignments. Then, they select a random customer and reassign it to a different valid vehicle and use customer insertion to adjust the route. Differently, other researchers [182] apply the greedy randomized adaptive search procedure (GRASP) meta-heuristic method. In particular, they initialize the assignments of orders based on some probabilities. In the group-assign framework, people [183, 184] optimally assign vehicles to shareable groups of customers. Generally, the group-assign framework achieves higher-quality assignments than the initialize-improve framework.

5.2 Order Dispatching

The target of order dispatching is to effectively and efficiently match taxi orders and vehicles. Generally speaking, existing work can be split into two types: rule-based and reinforcement-learning.

Rule-based approaches address the order dispatching problem by either centralized or decentralized ways. Lee et al. [185] and Lee et al. [186] implement the centralized method by the rule of “first-come, first-served.” Specifically, they regard the pick-up time/distance as the criterion and find the nearest option from a set of homogeneous drivers for each order. However, they ignore the potential optimal matching for each driver due to that there would be more suitable orders in the waiting list for a driver. To improve the global performance, Zhang et al. [187] combinatorially match multiple driver-order pairs within a short time window. Here, they distinguish different drivers by considering their long-term behavior history and short-term interests. For solving the problem in the decentralized setting, Seow et al. [188] divide drivers and orders into small groups and then simultaneously assign orders to driver within each group. Specifically, drivers conduct negotiations by several rounds of collaborative reasoning to decide whether to exchange current order assignments or not. However, it suffers from the limit of scalability due to the large communication cost among drivers. Alshamsi and Abdallah [189] also propose a system to support the negotiations between agents (drivers) to re-schedule allocated orders. In addition, they consider a sophisticated design of feature selection and weighting scheme as criteria to evaluate each driver-order pair.

Reinforcement-learning (RL) methods are recently popular for solving these sequential decision-making problems. Without additionally hand-crafted heuristics, they can learn an optimal policy based on observations and rewards provided by the environment. For example, Xu et al. [190] first propose an RL-based algorithm to dispatch resource in a

global and more farsighted view. However, it cannot better model interactions between multi-drivers and multi-orders due to its single-agent setting. In contrast, Li et al. [191] address the order dispatching problem using multi-agent reinforcement learning (MARL), which follows the distributed nature of the peer-to-peer ride sharing problem and possesses the ability to capture the stochastic demand-supply dynamics in large-scale ride sharing scenarios. Also, Jin et al. [192] build a multi-agent reinforcement learning framework, but they split the whole city into disjoint region cells and treat each region cell as an agent. To coordinate the agents from different regions to achieve long-term benefits, they leverage the geographical hierarchy of the region grids to perform hierarchical reinforcement learning.

5.3 Business Location

Select optimal locations to place retail store, charging station or billboard can increase business profit. Naturally, traffic data such as traffic flows and travel time play a significant role in solving this problem. For example, Karamshuk et al. [193] try to find the optimal placement of retail store with location-based social network. They explore how the popularity of retail store is shaped and conclude that the popularity is affected by the fusion of geographic and mobility features, which can be extracted from traffic flows. Therefore, predicting future traffic states would provide basic data for algorithms of selecting business locations.

As for solving the site selection of charging stations, aiming to reduce the detour distance, Li et al. [194] leverage historical trajectory data and spatial features of road network to design a deployment framework. In particular, they formalize the problem as an ILP (integer linear programming) optimization problem, which is NP-hard. Similarly, Liu et al. [195] convert the problem as a multiple-objective optimization problem, where they aim to maximize the overall revenue and minimize the overall driver discomfort.

Another significant business location problem is billboard placement, which aims to maximize the influence of billboards on passengers, also known as the influence maximization problem, where the influence is defined by the traffic flows. Guo et al. [196] focus on finding k buses, whose trajectories have maximum expected influence on audience, to deploy billboards. Liu et al. [197] try to select optimal placements (a vertex or edge who contains many traffic flows) on road networks to place outdoor billboards. Zhang et al. [198] consider the constraint of the total budget. They design a model on range and one-time impressions to solve the problem. However, their model has not considered the relationship between the influence effect and the impression counts for a single user. Hence, Zhang et al. [199] further propose a logistic influence model to address it. Wang et al. [200] also consider the constraint of budget, and they

use a divide-and-conquer strategy to improve the efficiency of placing billboards on road networks. Taking into account many factors (e.g., the customers' interest, the cooperation and competition among billboards) influencing the benefit of billboards, Lou et al. [201] formulate the dynamic advertising problem to maximize the commercial profit. More specifically, they first use the vehicular data (e.g., trajectories and preferences) to extract potential customers' implicit information. Then, they use the multi-agent deep reinforcement learning technique to propose an advertising strategy, by which the advertiser could determine the advertising policy for each billboard and maximize the commercial profit.

At last, some people focus on the general business location problem without any scenario. They try to select a set of facilities from the candidate set to maximize the influence with/without a cost budget. For example, Wang et al. [202] use the filtering-verification framework to prune many inferior candidate locations. Differently, Zhang et al. [203] formulate the problem as a geodemographic influence maximization problem, which is NP-hard. Hence, they propose a greedy algorithm with an approximation ratio.

5.4 Spatio-Temporal Anomaly Detection

Detecting spatio-temporal anomaly has been broadly studied. The target of this task is to identify the rare spatio-temporal data which are different from the majority. In other words, this task is one kind of classification problems. Hence, some traffic prediction techniques can be applied to solve some spatio-temporal based anomaly detection problems. In this paper, we focus on the task on three typical types of spatio-temporal data: event data, meteorological data and trajectory data.

Event data. Traffic conditions usually are influenced by casual events: such as car accidents, sports games and concerts. Thus, Sun et al. [204] have proposed a CNN-based model to detect the non-recurring traffic congestions caused by anomaly events. Also, Zhu et al. [205] use the CNN model to detect traffic accidents based on traffic flows. Differently, Zhang et al. [206] implement DBN (deep belief network) and LSTM models to detect event tweets related to traffic accidents based on social media data. At last, Chen et al. [207] study the relationship between traffic accidents and human mobility. In particular, they design a stack denoise autoencoder model to learn hierarchical feature representation of human mobility for predicting traffic accident risk level.

Meteorological data. At first, Liu et al. [208] utilize the deep learning model to detect climate extreme events, such as hurricanes and heat waves, based on climate image data. Also, Kim et al. [209] propose a framework to detect climate extreme events and reconstruct high-resolution climate data from the low-resolution climate data. In particular, they use

the CNN model to locally detect the extreme events, while designing a pixel recursive super-resolution model to recover coarse climate data. However, the number of extreme climate data is too sparse to train an effective model. Racah et al. [210] apply the semi-supervised model to improve the localization of extreme weathers. In particular, they present a multi-channel spatio-temporal CNN architecture by leveraging temporal information and unlabeled data.

Trajectory data. Detecting anomaly trajectory can help to identify criminal behaviors of taxi drivers [211, 212]. Generally, the anomaly trajectory from an origin point to a destination point is defined as the trajectory that appears with a low frequency. Many methods identify anomaly by computing the similarity among trajectories. In particular, they use the similarity to find the most dissimilar trajectories in a dataset. For example, Chen et al. [213] compute the similarity score between a given trajectory with existing trajectory that having the same origin and destination points in a dataset and then compare the score with a predefined threshold to determine whether the given trajectory is anomaly or not. Differently, Lee et al. [214] partition each trajectory into a set of line segments and then detect the anomaly by computing the similarity between different sets of line segments.

5.5 Route Planning

Route planning is one core component of intelligent transportation. Generally speaking, planning routes consist of two levels of tasks. On the one hand, we should be able to recommend proper tour routes for users. On the other hand, we can provide some suggestions to the construction of the transportation infrastructure. For example, we can help to plan bus routes or build new roads for relieve traffic congestion. Therefore, we survey existing work according to the above classification. By the way, both the two tasks rely on the prediction of some traffic states, such as traffic flows and travel time.

Tour route. The popular way to recommend tour routes is to find existing trips similar to given contexts, such as spatial proximity, text relevance and photographs. For example, Lu et al. [215] first leverage the geo-tagged photographs to recover travel clues and then recommend routes based on users' preference. In contrast, many people try to recommend popular routes. At first, Wei et al. [216] propose a search algorithm to find top-k popular trajectories, which pass through users' given regions. Later, Chen et al. [217] first leverage existing trips to build a tour network by linking hot areas with routes and then discover popular routes from the network with a traffic flow detection algorithm. At last, Wang et al. [218] implement an interactive route planning system, which can enable dynamic suggestion based on the click-based feedback from POIs displayed on the map.

Transportation infrastructure. Chen et al. [219] cluster all points of collected taxi trajectories and detect "hot spots" as recommended bus stops. In addition, they generate bus routes between any two stops with taxi trajectories. Also, Pinelli et al. [220] build transportation networks by computing traffic flows based on taxi trajectories. Differently, Wang et al. [221] leverage k nearest neighbor search method to find the route, whose distance is the least, to suggest the bus route of a given origin point and a given destination point. Hence, governments can build roads according to the networks. At last, Bao et al. [222] aim to suggest the building of bike lanes. In particular, they plan bile lines under the constraint of a budget and the number of connected components. In this paper, the authors propose a greedy network expansion algorithm, which can iteratively construct new lanes to reduce the number of connected components until the budget is met.

6 Emerging Challenges and Opportunities

In this section, we summarize some research challenges and opportunities in traffic prediction.

6.1 Complex Characteristics of Spatio-Temporal Data

Not only structured data but also unstructured data (e.g., pictures, texts, audios and videos) are used to predict traffic. For example, Liao et al. [171] consider the query information (text data) as auxiliary information when predicting traffic speed. Therefore, it requires to fuse multi-mode data. Audios and texts indicate the sequential characteristic, so we can use sequence encoding techniques (e.g., RNN and attention) to learn or extract their features. Pictures and videos have be handled in the domain of compute vision by the CNN technique, so we can apply it to related traffic data. At last, some social media data, such as geo-tagged twitters, have influence on the traffic prediction, and we can utilize the graph-based models (e.g., GNN) to learn or extract related features, due to the graph structure of social network.

Collected data are often unevenly distributed. For example, there exit dense traffic on some roads, while others are sparse, which would cause the difficulty of sparse traffic prediction due to the lack of training data. To address this issue, the possible way is to adopt some advanced techniques, such as zero/few-shot learning and meta learning.

6.2 AI-enhanced Spatio-Temporal Data Preprocessing

It becomes popular to utilize AI techniques to enhance data-based managements [223–244]. Naturally, these techniques

can be transferred to help the management of spatio-temporal data. (a) It is inefficient to clean data with handcraft rules, so we can design different learning models to address different data problems. (b) Similar to [245] and [246], which build learned indexes to accelerate the query on large scale of multi-dimensional data, we can use learned indexes to improve the distributed storage of spatio-temporal data. (c) Data compression can be regarded as the generative problem, so we can use generative learning models (e.g., VAE and GAN) to address this problem.

6.3 Joint Traffic Prediction

Most existing work proposes different models to solve different types of traffic prediction problems. Although they consider various features, such as the spatio-temporal properties and environmental data, the relationship between different types of traffic data has not been significantly used. For example, as claimed in [247], if there are increasing travel demands in a region, the traffic flows in the region would also increase in a near future. Hence, we need to handle the traffic prediction problem by jointly considering different types of traffic data. Also, the opportunity of improving the performance of traffic prediction is to address the challenge of joint traffic prediction. The challenge is twofold. On the one hand, different types of data correspond to different formats, so we need to address the issue that different formats should be fused. On the other hand, the influence or relationship between different types of traffic data is asymmetric, so how to model it becomes difficult.

6.4 Interpretable and Automatic Deep Traffic Prediction Models

As described in Sect. 4, many traffic prediction models are implemented with deep learning techniques. However, most of these models just like “black-box” for getting prediction results. In contrast, making decision on the building of intelligent transportation should depend on reasonability and interpretability of traffic prediction results. It is thereby significant to design interpretable deep learning models. In addition, training a deep learning model is always expensive due to the heavy exploration of hyper-parameters in models. Therefore, how to automatically design effective and efficient models would be a significant topic in the traffic prediction community.

6.5 Unified Intelligent Transportation System

The final target of traffic prediction is to make real transportation intelligent. In other words, we could gain convenient travel services no matter when and where we need. To achieve this goal, we need to build an unified intelligent

transportation system, which can manage, analyze and mining all spatio-temporal data. However, there exist some challenges. (a) How to make different data sources be trust, because we need different organizations or companies to share their data to the unified system. The opportunity is that some useful techniques (e.g., federal learning) seem to be useful. (b) It is expensive to handle the change of online traffic, especially the update of maps. Therefore, it is a big challenge to guarantee the efficiency of associated services.

6.6 Performance Benchmarks and Pre-train Models

Notably, most studies related to traffic prediction just build task-oriented datasets, such as trajectories. However, urban traffic data include many complex factors or features. Hence, how to construct a completed and unified dataset is significant for the development of the traffic prediction. In addition, the essential operation of most learning-based methods for traffic prediction is to learn the vector representation of spatio-temporal data. Therefore, similar to the pre-training model of representation learning in the field of NLP (Natural Language Processing), such as BERT [248] and GPT-3 [249], we can also pre-train a general model to represent spatio-temporal data.

7 Public Spatio-Temporal Datasets

Thanks to some enterprises and researchers in this field, there are quite a few real spatio-temporal datasets that are publicly available:

- *GAIA Open Dataset*³: Didi provides academic community with real-life, high-quality anonymized data. In the website, they provide not only raw order-related datasets (e.g., orders, trajectories and voice data), but also self-processing transportation index datasets (e.g., travel time index and transportation energy index). In addition, they build benchmark datasets for some popular transportation data mining competitions, such as KDD CUP 2020⁴ and CCF BDCI 2020⁵.
- *Open Street Map (OSM)*⁶: Road networks are broadly applied in many traffic prediction problems. OSM provides the way to access the road network all over the world. Also, we can extract the road network for each special city.

³ <https://outreach.didichuxing.com/research/opendata/>.

⁴ <https://www.kdd.org/kdd2020/kdd-cup>.

⁵ <https://www.datafountain.cn/special/BDCI2020>.

⁶ <https://www.openstreetmap.org/>.

- *Taxi Trajectories*: There are plenty of taxi trajectories released from some research projects. For example, Yuan et al. [250] provide a dataset, which is a sample of trajectories from Microsoft Research T-Drive project, generated by over 10,000 taxicabs in a week of 2008 in Beijing. In addition, the taxi service trajectory prediction challenge 2015⁷ also provides an accurate dataset describing complete year (from 01/07/2013 to 30/06/2014) of the (busy) trajectories performed by all the 442 taxis running in the city of Porto.

8 Conclusion

In this paper, we review extensive studies on traffic prediction. In particular, these studies run from the spatio-temporal data layer to the intelligent transportation application layer. We first summarize the traffic prediction use cases and then propose the overview of traffic prediction, which includes four parts: spatio-temporal data, preprocessing, traffic prediction and traffic application. First, we review different types of traffic data. Second, we survey all of existing work on how to preprocess these traffic data. Third, we summarize the challenges for traffic prediction and also survey all of existing techniques about addressing these challenges. Fourth, we discuss how to implement traffic applications to make the transportation intelligent. Finally, we provide emerging challenges and opportunities.

Funding This paper was supported by NSF of China (61925205, 61632016), Huawei, TAL education, and Beijing National Research Center for Information Science and Technology (BNRist).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Xie P, Li T, Liu J, Du S, Yang X, Zhang J (2020) Urban flow prediction from spatiotemporal data using machine learning: a survey. *Inf Fusion* 59:1–12
2. Zheng Y (2019) *Urban computing*. MIT Press, Cambridge
3. Rabiner L, Juang B (1986) An introduction to hidden markov models. *IEEE ASSP Mag* 3(1):4–16
4. Lafferty J, McCallum A, Pereira Fernando CN (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: *ICML*, pp 282–289
5. Ross Quinlan J (1986) Induction of decision trees. *Mach Learn* 1(1):81–106
6. Steve Lawrence C, Lee Giles A, Tsoi C, Back AD (1997) Face recognition: a convolutional neural-network approach. *IEEE Trans Neural Netw* 8(1):98–113
7. Medsker L, Jain LC (1999) *Recurrent neural networks: design and applications*. CRC Press, Boca Raton
8. Stone JV (2013) *Bayes' rule: A tutorial introduction to Bayesian analysis*. Sebtel Press, England
9. Srivastava T (2014) Introduction to knn, k-nearest neighbors: simplified. *Analytics Vidhya, Gurgaon*
10. Alireza M, Jonathon S, Navdeep J, Ian G, Brendan F (2015) Adversarial autoencoders. *arXiv:1511.05644*
11. Tedjopurnomo DA, Bao Z, Zheng B, Choudhury F, Qin AK (2020) A survey on modern deep neural network for traffic prediction: trends, methods and challenges. *TKDE*, pp 1–1
12. Barros J, Araujo M, Rossetti Rosaldo JF (2015) Short-term real-time traffic prediction methods: a survey. In: *MT-ITS*. *IEEE*, pp. 132–139
13. Li Y, Shahabi C (2018) A brief overview of machine learning methods for short-term traffic forecasting and future directions. *SIGSPATIAL Spec* 10(1):3–9
14. Nagy AM, Simon V (2018) Survey on traffic prediction in smart cities. *Pervasive Mobile Comput* 50:148–163
15. Shi X, Yeung D-Y (2018) Machine learning for spatiotemporal sequence forecasting: a survey. *arXiv:1808.06865*
16. Shi Y, Feng H, Geng X, Tang X, Wang Y (2019) A survey of hybrid deep learning methods for traffic flow prediction. In: *Proceedings of the 2019 3rd international conference on advances in image processing*, pp 133–138
17. Wang S, Cao J, Yu PS (2019) Deep learning for spatio-temporal data mining: a survey. *arXiv:1906.04928*
18. Wang S, Bao Z, Culpepper JS, Cong G (2020) A survey on trajectory data management, analytics, and learning. *arXiv:2003.11547*
19. Tang J, Zheng L, Han C, Yin W, Zhang Y, Zou Y, Huang H (2020) Statistical and machine-learning methods for clearance time prediction of road incidents: a methodology review. In: *Analytic Methods in Accident Research*, vol. 27, pp 1–16
20. Tong Y, Chen Y, Zhou Z, Chen L, Wang J, Yang Q, Ye J, Lv W (2017) The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In: *SIGKDD*, pp. 1653–1662
21. Yao H, Wu F, Ke J, Tang X, Jia Y, Lu S, Gong P, Ye J, Li Z (2018) Deep multi-view spatial-temporal network for taxi demand prediction. In: *AAAI*, pp 2588–2595
22. Zheng Y, Capra L, Wolfson O, Yang H (2014) Urban computing: concepts, methodologies, and applications. *TIST* 5(3):38:1–38:55
23. Zheng Y (2015) Trajectory data mining: an overview. *TIST* 6(3):29:1–29:41
24. Jia Z, Chen C, Coifman B, Varaiya P (2001) The pems algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors. In: *ITSC*, pp 536–541
25. Petty KF, Bickel P et al (1998) Accurate estimation of travel times from single-loop detectors. *Transp Res Part A* 32(1):1–17
26. Tang J, Zou Y, Ash J, Zhang S, Liu F, Wang Y (2016) Travel time estimation using freeway point detector data based on evolving fuzzy neural inference system. *PLoS ONE* 11(2):1–24
27. Ding Z, Yang B, Güting RH, Li Y (2015) Network-matched trajectory-based moving-object database: models and applications. *TITS* 16(4):1918–1928

⁷ <http://www.geolink.pt/ecmlpkdd2015-challenge/>.

28. Hunter T, Herring R, Abbeel P, Bayen A (2009) Path and travel time inference from gps probe vehicle data. *NIPS* 12(1):2
29. Atluri G, Karpatne A, Kumar V (2018) Spatio-temporal data mining: a survey of problems and methods. *Comput Surv* 51(4):83:1–83:41
30. Zhao K, Yang YH, Qu BZ (2003) Gps/dr group and the navigation system map matching algorithm based on position points matching. *Guidance Fuses* 24(3):22–27
31. Greenfeld Joshua S (2002) Matching gps observations to locations on a digital map. In: 81th annual meeting of the transportation research board, vol 1., Washington DC, pp 164–173
32. Mohammed Q, Simon W (2015) Shortest path and vehicle trajectory aided map-matching for low frequency gps data. *Transp Res Part C Emerg Technol* 55:328–339
33. Chawathe Sudarshan S (2007) Segment-based map matching. In: *IVS*, pp 1190–1197
34. Civilis A, Jensen CS, Pakalnis S (2005) Techniques for efficient road-network-based tracking of moving objects. *TKDE* 17(5):698–712
35. Hong W, Yin W, George F, Yanmin Z (2013) Map matching by fréchet distance and global weight optimization. *Departement of Computer Science and Engineering, Technical Paper*, p 19
36. Zhu L, Holden JR, Gonder JD (2017) Trajectory segmentation map-matching approach for large-scale, high-resolution gps data. *Transp Res Rec* 2645(1):67–75
37. Zheng K, Zheng Y, Xie X, Zhou X (2012) Reducing uncertainty of low-sampling-rate trajectories. In: *ICDE*, pp 1144–1155
38. Alt H, Efrat A, Rote G, Wenk C (2003) Matching planar maps. *J Algorithms* 49(2):262–283
39. Brakatsoulas S, Pfoer D, Salas R, Wenk C (2005) On map-matching vehicle tracking data. In: *Proceedings of the 31st international conference on Very large data bases*, pp 853–864
40. Chen W, Yu M, Li ZL, Chen YQ (2003) Integrated vehicle navigation system for urban applications. In: *GNSS*, pp 15–22
41. Washington YO, Mohammed Q, Robert BN (2003) Map-matching in complex urban road networks. *Revista Brasileira de Cartografia* 55(2):1–14
42. Quddus MA, Noland RB, Ochieng WY (2006) A high accuracy fuzzy logic based map matching algorithm for road transport. *J Intell Trans Syst* 10(3):103–115
43. Pink O, Hummel B (2008) A statistical approach to map matching using road network geometry, topology and vehicular motion constraints. In: *ICITS*, pp 862–867
44. Lou Y, Zhang C, Zheng Y, Xie X, Wang W, Huang Y (2009) Map-matching for low-sampling-rate gps trajectories. In: *SIGSPATIAL*, pp 352–361
45. Newson P, Krumm J (2009) Hidden markov map matching through noise and sparseness. In: *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pp 336–343
46. Yuan J, Zheng Y, Zhang C, Xie X, Sun G-Z (2010) An interactive-voting based map matching algorithm. In: *MDM*, pp 43–52
47. Bonnifait P, Laneurit J, Fouque C, Dherbomez G (2009) Multi-hypothesis map-matching using particle filtering. *16th World Congress for ITS Systems and Services*. Stockholm, Sweden, pp 1–8
48. Chong YG, Justin D, Nikola M, Muhammad TA, Ali O, Patrick J (2012) Online map-matching based on hidden markov model for real-time traffic sensing applications. In: *ITSC*, pp 776–781
49. Wang X, Ni W (2016) An improved particle filter and its application to an ins/gps integrated navigation system in a serious noisy scenario. *Meas Sci Technol* 27(9):095005
50. Takayuki O, Rudy R (2013) Map matching with inverse reinforcement learning. In: *IJCAI*, pp 2547–2553
51. Hunter T, Abbeel P, Bayen AM (2014) The path inference filter: model-based low-latency map matching of probe vehicle data. *TITS* 15(2):507–529
52. Gang H, Shao J, Liu F, Wang Y, Shen HT (2017) If-matching: towards accurate map-matching with information fusion. *TKDE* 29(1):114–127
53. Sharath MN, Velaga NR, Quddus MA (2019) A dynamic two-dimensional (d2d) weight-based map-matching algorithm. *Transp Res Part C Emerg Technol* 98:409–432
54. Zhao K, Feng J, Xu Z, Xia T, Chen L, Sun F, Guo D, Jin D, Li Y (2019) Deepmm: deep learning based map matching with data augmentation. In: *SIGSPATIAL*, pp 452–455
55. Xi L, Liu Q, Li M, Liu Z (2007) Map matching algorithm and its application. In: *International conference on intelligent systems and knowledge engineering 2007*. Atlantis Press
56. Chao P, Xu Y, Hua W, Zhou X (2020) A survey on map-matching algorithms. In: *ADC*, pp 121–133
57. Lee D, Kulic D, Nakamura Y (2008) Missing motion data recovery using factorial hidden markov models. In: *ICRA*, pp 1722–1728
58. Yi X, Zheng Y, Zhang J, Li T (2016) St-mvl: filling missing values in geo-sensory time series data. In: *IJCAI*, pp. 2704–2710
59. Lin Z, Chen M, Ma Y (2010) The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *CoRR*, abs/1009.5055
60. Ruan W, Xu P, Sheng QZ, Falkner NJG, Li X, Zhang WE (2017) Recovering missing values from corrupted spatio-temporal sensory data via robust low-rank tensor completion. In: *DSFAA*, pp 607–622
61. Knorr Edwin M, Ng Raymond T, Vladimir T (2000) Distance-based outliers: algorithms and applications. *VLDB J* 8(3–4):237–253
62. Lu C-T, Chen Dechang, Kou Yufeng (2003) Algorithms for spatial outlier detection. In: *ICDM*, pp 597–600
63. Shekhar S, Lu C-T, Zhang P (2001) Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In: *SIGKDD*, pp 371–376
64. Kou Yufeng L, Chang-Tien DS, Raimundo F (2007) Spatial outlier detection: a graph-based approach. *ICTAI* 1:281–288
65. Kut A, Birant D (2006) Spatio-temporal outlier detection in large databases. *J Comput Inf Technol* 14(4):291–297
66. Mauder M, Reisinger M, Emrich T, Züfle A, Renz M, Trajcevski G, Tamassia R (2015) Minimal spatio-temporal database repairs. In: *ISSTD*, pp 255–273
67. Zhou H, Zhang D, Xie K, Chen Y (2016) Robust spatio-temporal tensor recovery for internet traffic data. In: *BigDataSE/ISPA*, pp 1404–1411
68. Zheng Y, Liu F, Hsieh H-P (2013) U-air: when urban air quality inference meets big data. In: *SIGKDD*, pp 1436–1444
69. Beckmann M, Ebecken NFF, Beatriz SL, de Lima P et al (2015) A knn undersampling approach for data balancing. *J Intell Learn Syst Appl* 7(04):104
70. Wang R, Kwong S, Jia Y, Huang Z, Wu L (2018) Mutual information based k-labelsets ensemble for multi-label classification. In: *FUZZ-IEEE*, pp 1–7
71. Gong J, Kim H (2017) Rhsboost: improving classification performance in imbalance data. *Comput Stat Data Anal* 111:1–13
72. Guttman A (1984) R-trees: a dynamic index structure for spatial searching. In: *SIGMOD*, pp 47–57
73. Pfoer D, Jensen CS, Theodoridis Y, et al. (2000) Novel approaches to the indexing of moving object trajectories. In: *VLDB*, pp 395–406
74. Xu X, Han J, Lu W (1990) Rt-tree: an improved r-tree indexing structure for temporal spatial databases. In: *SDH*, pp 1040–1049

75. Tao Y, Papadias D (2001) Efficient historical r-trees. In: SSDBM, pp 223–232
76. Tao Y, Papadias D (2001) The mv3r-tree: a spatio-temporal access method for timestamp and interval queries. In: VLDB
77. Nievergelt J, Hinterberger H, Sevcik KC (1984) The grid file: an adaptable, symmetric multikey file structure. *TODS* 9(1):38–71
78. Wang L, Zheng Y, Xie X, Ma W-Y (2008) A flexible spatio-temporal indexing scheme for large-scale gps track retrieval. In: MDM, pp 1–8
79. Zhong R, Li G, Tan K-L, Zhou L, Gong Z (2015) G-tree: an efficient and scalable index for spatial search on road networks. *TKDE* 27(8):2175–2189
80. Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I et al (2010) Spark: cluster computing with working sets. *HotCloud* 10(10–10):95
81. Eldawy A, Mokbel MF (2015) Spatialhadoop: a mapreduce framework for spatial data. In: ICDE, pp 1352–1363
82. Aji A, Wang F, Vo H, Lee R, Liu Q, Zhang X, Saltz J (2013) Hadoop-gis: a high performance spatial data warehousing system over mapreduce. In: VLDB, volume 6. NIH Public Access
83. Tan H, Luo W, Ni LM (2012) Clost: a hadoop-based storage system for big spatio-temporal data analytics. In: CIKM, pp 2139–2143
84. Yu J, Wu J, Sarwat M (2015) Geospark: a cluster computing framework for processing large-scale spatial data. In: SIGSPATIAL, pp 1–4
85. Xie D, Li F, Yao B, Li G, Zhou L, Guo M (2016) Simba: efficient in-memory spatial analytics. In: SIGMOD, pp 1071–1085
86. Armbrust M, Xin RS, Lian C, Huai Y, Liu D, Bradley JK, Meng X, Kaftan T, Franklin MJ, Ghodsi A, et al (2015) Spark sql: relational data processing in spark. In: SIGMOD, pp 1383–1394
87. Xie D, Li F, Phillips JM (2017) Distributed trajectory similarity search. In: VLDB, pp 1478–1489
88. Shang Z, Li G, Bao Z (2018) Dita: distributed in-memory trajectory analytics. In: SIGMOD, pp 725–740
89. Yuan H, Li G (2019) Distributed in-memory trajectory similarity search and join on road network. In: ICDE, pp 1262–1273
90. Douglas DH, Peucker TK (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartogr Int J Geogr Inf Geovis* 10(2):112–122
91. Chen Y, Jiang K, Zheng Y, Li C, Yu N (2009) Trajectory simplification method for location-based social networking services. In: LBSN, pp 33–40
92. Zhang D, Ding M, Yang D, Yi Liu J, Fan, and Heng Tao Shen, (2018) Trajectory simplification: an experimental study and quality analysis. *VLDB* 11(9):934–946
93. Song R, Sun W, Zheng B, Zheng Y (2014) Press: a novel framework of trajectory compression in road networks. *arXiv:1402.1546*
94. Yang X, Wang B, Yang K, Liu C, Zheng B (2017) A novel representation and compression for queries on trajectories in road networks. *TKDE* 30(4):613–629
95. Keogh E, Chu S, Hart D, Pazzani M (2001) An online algorithm for segmenting time series. In: ICDM, pp 289–296
96. Meratnia N, Rolf A (2004) Spatiotemporal compression techniques for moving point objects. In: EDBT, pp 765–782
97. Potamias M, Patroumpas K, Sellis T (2006) Sampling trajectory streams with spatiotemporal criteria. In: SSDBM, pp 275–284
98. Krumm J, Horvitz E (2004) Locadio: inferring motion and location from wi-fi signal strengths. In: *Mobiquitous*, pp 4–13
99. Sohn T, Varshavsky A, LaMarca A, Chen MY, Choudhury T, Smith I, Consolvo S, Hightower J, Griswold WG, De Lara E (2006) Mobility detection using everyday gsm traces. In: ICUC, pp 212–224
100. Zhu Y, Zheng Y, Zhang L, Santani D, Xie X, Yang Q (2012) Inferring taxi status using gps trajectories. *arXiv:1205.4378*
101. Zheng Y, Li Q, Chen Y, Xie X, Ma W-Y (2008) Understanding mobility based on gps data. In: ICUC, pp 312–321
102. Zheng Y, Liu L, Wang L, Xie X (2008) Learning transportation mode from raw gps data for geographic applications on the web. In: WWW, pp 247–256
103. Liao L, Patterson DJ, Fox D, Kautz H (2007) Learning and inferring transportation routines. *Artif Intell* 171(5–6):311–331
104. Patterson DJ, Liao L, Fox D, Kautz H (2003) Inferring high-level behavior from low-level sensors. In: ICUC, pp 73–89
105. Yin J, Chai X, Yang Q (2004) High-level goal recognition in a wireless lan. In: AAAI, pp 578–584
106. Stenneth L, Wolfson O, Yu PS, Xu B (2011) Transportation mode detection using mobile phones and gis information. In: SIGSPATIAL, pp 54–63
107. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: *NeurIPS*, pp 1097–1105
108. Nolte M, Kister N, Maurer M (2018) Assessment of deep convolutional neural networks for road surface classification. In: ITSC, pp 381–386
109. Ramanna S, Sengoz C, Kehler S, Pham D (2020) Near real-time map building with multi-class image set labelling and classification of road conditions using convolutional neural networks. *arXiv:2001.09947*
110. Pamula T (2018) Road traffic conditions classification based on multilevel filtering of image content using convolutional neural networks. *ITSM* 10(3):11–21
111. Liu H, Lee I (2017) End-to-end trajectory transportation mode classification using bi-lstm recurrent neural network. In: ISKE, pp 1–5
112. Yanjun Q, Haiyong L, Fang Z, Chenxing W, Jiaqi W, Yuexia Z (2019) Toward transportation mode recognition using deep convolutional and long short-term memory recurrent neural networks. *IEEE Access* 7:142353–142367
113. Nawaz A, Zhiqiu H, Senzhang W, Hussain Y, Khan I, Khan Z (2020) Convolutional lstm based transportation mode learning from raw gps trajectories. *ITS* 14(6):570–577
114. Wang Chenxing, Luo Haiyong, Zhao Fang, Qin Yanjun (2020) Combining residual and lstm recurrent networks for transportation mode detection using multimodal sensors integrated in smartphones. *TITS*
115. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: ECCV, pp. 630–645
116. Liu H, Wu H, Sun W, Lee I (2019) Spatio-temporal gru for trajectory classification. In: ICDM, pp 1228–1233
117. Brinkhoff T (2000) Generating network-based moving objects. In: *Proceedings 12th international conference on scientific and statistica database management*, pp 253–255
118. Daniel K, Jakob E, Michael B, Laura B (2012) Recent development and applications of sumo-simulation of urban mobility. *Int J Adv Syst Meas* 5:128–138
119. van Lon Rinde RS, Holvoet T (2012) Rinsim: a simulator for collective adaptive systems in transportation and logistics. In: SASO, pp 231–232
120. Adnan M, Pereira FC, Azevedo CML, Basak K, Lovric M, Raveau S, Zhu Y, Ferreira J, Zegras C, Ben-Akiva M (2016) Simmobility: a multi-scale integrated agent-based simulation platform. In: *Trans Res Board 95th Annual Meeting*
121. Pan JJ, Li G, Juntao H (2019) Ridesharing: simulator, benchmark, and evaluation. *VLDB* 12(10):1085–1098
122. Hongjian W, Yu-Hsuan K, Daniel K, Zhenhui L (2016) A simple baseline for travel time estimation using large-scale trip data. *SIGSPATIAL*, pp 61(1–61):4

123. Jindal I, Chen X, Nokleby M, Ye J, et al (2017) A unified neural network approach for estimating travel time and distance for a taxi trip. *CoRR*
124. Li Y, Fu K, Wang Z, Shahabi C, Ye J, Liu Y (2018) Multi-task representation learning for travel time estimation. In: *SIGKDD*, pp 1695–1704
125. Yuan H, Li G, Bao Z, Feng L (2020) Effective travel time estimation: when historical trajectories over road networks matter. In: *SIGMOD*, pp 2135–2149
126. Rahmani M, Jenelius E, Koutsopoulos HN (2013) Route travel time estimation using low-frequency floating car data. In: *ITSC*, pp 2292–2297
127. Wang Y, Zheng Y, Xue Y (2014) Travel time estimation of a path using sparse trajectories. In: *KDD*, pp 25–34
128. Idé T, Sugiyama M (2011) Trajectory regression on road networks. In: *AAAI*, pp 203–208
129. Zheng J, Ni LM (2013) Time-dependent trajectory regression on road networks via multi-task learning. In: *AAAI*, pp 1048–1055
130. Gal A, Mandelbaum A, Schnitzler F, Senderovich A, Weidlich M (2017) Traveling time prediction in scheduled transportation with journey segments. *Inf Syst* 64:266–280
131. Yang B, Guo C, Jensen CS (2013) Travel cost inference from sparse, spatio-temporally correlated time series using markov models. *VLDB* 6(9):769–780
132. Dong W, Junbo Z, Wei C, Jian L, Yu Z (2018) When will you arrive? Estimating travel time based on deep neural networks. In: *AAAI*, pp 2500–2507
133. Zhang H, Wu H, Sun W, Zheng B (2018) Deepttravel: a neural network based travel time estimation model with auxiliary supervision. In: *IJCAI*, pp 3655–3661
134. Wang Z, Fu K, Ye J (2018) Learning to estimate the travel time. In: *SIGKDD*, pp 858–866
135. Asghari M, Emrich T, Demiryurek U, Shahabi C (2015) Probabilistic estimation of link travel times in dynamic road networks. In: *SIGSPATIAL*, pp 1–10
136. Dai J, Yang B, Guo C, Jensen CS, Jilin H (2016) Path cost distribution estimation using trajectory data. *VLDB* 10(3):85–96
137. Li X, Cong G, Sun A, Cheng Y (2019) Learning travel time distributions with deep generative model. In: *WWW*, pp 1017–1027
138. Cryer JD (1986) Time series analysis, volume 286
139. Saadallah A, Moreira-Matias L, Sousa R, Khiari J, Jenelius E, Gama J (2018) Bright-drift-aware demand predictions for taxi networks. *TKDE*
140. Wang D, Cao W, Li J, Ye J (2017) Deepdsd: supply-demand prediction for online car-hailing services using deep neural networks. In: *ICDE*, pp 243–254
141. Bai L, Yao L, Kanhere SS, Yang Z, Chu J, Wang X (2019) Passenger demand forecasting with multi-task convolutional recurrent neural networks. In: *PAKDD*, pp 29–42
142. Kuang L, Yan X, Tan X, Li S, Yang X (2019) Predicting taxi demand based on 3d convolutional neural network and multi-task learning. *Remote Sens* 11(11):1265
143. Liu L, Qiu Z, Li G, Wang Q, Ouyang W, Lin L (2019) Contextualized spatial-temporal network for taxi origin-destination demand prediction. *TITS* 20(10):3875–3887
144. Chu K-F, Lam AYS, Li VO Deep multi-scale convolutional lstm network for travel demand and origin-destination predictions. *TITS*, pp 1–14
145. Wu W, Liu T, Yang J (2020) Cacrrn: a context-aware attention-based convolutional recurrent neural network for fine-grained taxi demand prediction. In: *PAKDD*, pp 636–648
146. Geng X, Yaguang L, Leye W, Lingyu Z, Qiang Y, Jieping Y, Yan L (2019) Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. *AAAI* 33:3656–3663
147. Ying X, Li D (2019) Incorporating graph attention and recurrent architectures for city-wide taxi demand prediction. *Geo-Inf* 8(9):414
148. Bai L, Yao L, Kanhere SS, Wang X, Sheng QZ (2019) Stg2seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting. In: *IJCAI*, pp 1981–1987
149. Chu J, Qian K, Wang X, Yao L, Xiao F, Li J, Miao X, Yang Z (2018) Passenger demand prediction with cellular footprints. In: *SECON*, pp 1–9
150. Guy L, Yaacov R (2007) Traffic flow prediction using adaboost algorithm with random forests as a weak learner. *Proc World Acad Sci Eng Technol* 19:193–198
151. Zhang J, Zheng Y, Qi D (2017) Deep spatio-temporal residual networks for citywide crowd flows prediction. In: *AAAI*, pp 1655–1661
152. He Z, Chow C-Y, Zhang J-D (2019) Stcnn: a spatio-temporal convolutional neural network for long-term traffic prediction. In: *MDM*, pp 226–233
153. Huaxiu Y, Xianfeng T, Hua W, Guanjie Z, Zhenhui L (2019) Revisiting spatial-temporal similarity: a deep learning framework for traffic prediction. *AAAI* 33:5668–5675
154. Jin X, Zhang Y, Yao D (2007) Simultaneously prediction of network traffic flow based on PCA-SVR. In: *ISNN*, pp 1022–1031
155. Jinjun T, Chen Xinqiang H, Zheng ZF, Chunyang H, Leixiao L (2019) Traffic flow prediction based on combination of support vector machine and data denoising schemes. *Phys A Stat Mech Appl* 534:120642
156. Tang J, Gao F, Liu F, Chen X (2020) A denoising scheme-based traffic flow prediction model: combination of ensemble empirical mode decomposition and fuzzy c-means neural network. *IEEE Access* 8:11546–11559
157. Yan Y, Zhang S, Tang J, Wang X (2017) Understanding characteristics in multivariate traffic flow time series from complex network structure. *Phys A Stat Mech Appl* 477:149–160
158. Lv Y, Duan Y, Kang W, Li Z, Wang F-Y (2014) Traffic flow prediction with big data: a deep learning approach. *TITS* 16(2):865–873
159. Fang S, Zhang Q, Meng G, Xiang S, Pan C (2019) Gstnet: global spatial-temporal network for traffic flow prediction. In: *IJCAI*, pp 10–16
160. Wang M, Lai B, Jin Z, Lin Y, Gong X, Huang J, Hua X (2018) Dynamic spatio-temporal graph-based cnns for traffic prediction. *arXiv:1812.02019*
161. Shengnan G, Youfang L, Ning F, Chao S, Huaiyu W (2019) Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. *AAAI* 33:922–929
162. Li Y, Yu R, Shahabi C, Liu Y (2018) Diffusion convolutional recurrent neural network: data-driven traffic forecasting. *ICLR(Poster)*
163. Wang X, Ma Y, Wang Y, Jin W, Wang X, Tang J, Jia C, Yu J (2020) Traffic flow prediction via spatial temporal graph neural network. In: *WWW*, pp 1082–1092
164. Pan Z, Liang Y, Wang W, Yu Y, Zheng Y, Zhang J (2019) Urban traffic prediction from spatio-temporal data using deep meta learning. In: *SIGKDD*, pp 1720–1730
165. Lv Z, Xu J, Zheng K, Yin H, Zhao P, Zhou X (2018) Lc-rnn: a deep learning model for traffic speed prediction. In: *IJCAI*, pp 3470–3476
166. Ma X, Dai Z, He Z, Ma J, Wang Y, Wang Y (2017) Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* 17(4):818
167. Wang W, Li X (2018) Travel speed prediction with a hierarchical convolutional neural network and long short-term memory model framework. *arXiv:1809.01887*

168. Cui Z, Ke R, Pu Z, Wang Y (2018) Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction. *arXiv:1801.02143*
169. Tang J, Liu F, Zou Y, Zhang W, Wang Y (2017) An improved fuzzy neural network for traffic speed prediction considering periodic characteristic. *TITS* 18(9):2340–2350
170. Yang X, Zou Y, Tang J, Liang J, Ijaz M (2020) Evaluation of short-term freeway speed prediction based on periodic analysis using statistical models and machine learning models. *J Adv Trans* 1–16:2020
171. Liao B, Zhang J, Wu C, McIlwraith D, Chen T, Yang S, Guo Y, Wu F (2018) Deep sequence learning with auxiliary information for traffic prediction. In: *SIGKDD*, pp 537–546
172. Song HY, Baek MS, Sung M (2019) Generating human mobility route based on generative adversarial network. In: *FedCSIS*, pp 91–99
173. Wu H, Chen Z, Sun W, Zheng B, Wang W (2017) Modeling trajectories with recurrent neural networks. In: *IJCAI*
174. Cordeau J-F (2006) A branch-and-cut algorithm for the dial-a-ride problem. *Oper Res* 54(3):573–586
175. Tian C, Huang Y, Liu Z, Bastani F, Jin R (2013) Noah: a dynamic ridesharing system. In: *SIGMOD*, pp 985–988
176. Jaeyoung Jung R, Jayakrishnan, Park JY (2016) Dynamic shared-taxi dispatch algorithm with hybrid-simulated annealing. *Comput-Aid Civ Infrastruct Eng* 31(4):275–291
177. Ma S, Zheng Y, Wolfson O (2013) T-share: a large-scale dynamic taxi ridesharing service. In: *ICDE*, pp 410–421
178. Jaw J-J, Odoni AR, Psaraftis HN, Wilson NHM (1986) A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transp Res Part B Methodol* 20(3):243–257
179. Shuo M, Zheng Y, Ouri W (2014) Real-time city-scale taxi ridesharing. *TKDE* 27(7):1782–1795
180. Yan H, Favyen B, Ruoming J, Xiaoyang SW (2014) Large scale real-time ridesharing with service guarantee on road networks. *VLDB*, 7(14)
181. Cheng P, Xin H, Chen L (2017) Utility-aware ridesharing on road networks. In: *SIGMOD*, pp 1197–1210
182. Santos DO, Xavier EC (2013) Dynamic taxi and ridesharing, a framework and heuristics for the optimization problem. In: *IJCAI*
183. Alonso-Mora J, Samaranayake S, Wallar A, Frazzoli E, Rus D (2017) On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *PNAS* 114(3):462–467
184. Ta N, Li G, Zhao T, Feng J, Ma H, Gong Z (2017) An efficient ride-sharing framework for maximizing shared route. *IEEE Trans Knowl Data Eng* 30(2):219–233
185. Lee D-H, Wang H, Ruey LC, Siew HT (2004) Taxi dispatch system based on current demands and real-time traffic conditions. *Transp Res Rec* 1:193–200
186. Lee J, Park G-L, Kim H, Yang Y-K, Kim P, Kim S-W (2007) A telematics service system based on the linux cluster. In: *ICCS*, pp 660–667
187. Zhang L, Hu T, Min Y, Wu G, Zhang J, Feng P, Gong P, Ye J (2017) A taxi order dispatch model based on combinatorial optimization. In: *SIGKDD*, pp 2151–2159
188. Seow KT, Dang NH, Lee D-H (2009) A collaborative multiagent taxi-dispatch system. *TASE* 7(3):607–616
189. Alshamsi A, Abdallah S, Rahwan I (2009) Multiagent self-organization for a taxi dispatch system. In: *ICAAMS*, pp 21–28
190. Xu Z, Li Z, Guan Q, Zhang D, Li Q, Nan J, Liu C, Bian W, Ye J (2018) Large-scale order dispatch in on-demand ride-hailing platforms: a learning and planning approach. In: *SIGKDD*, pp 905–913
191. Li M, Qin Z, Jiao Y, Yang Y, Wang J, Wang C, Wu G, Ye J (2019) Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In: *WWW*, pp 983–994
192. Jiarui J, Ming Z, Weinan Z, Minne L, Zilong G, Zhiwei Q, Yan J, Xiaocheng T, Chenxi W, Jun W et al. (2019) Coride: joint order dispatching and fleet management for multi-scale ride-hailing platforms. *CIKM* pp 1983–1992
193. Karamshuk D, Noulas A, Scellato S, Nicosia V, Mascolo C (2013) Geo-spotting: mining online location-based services for optimal retail store placement. In: *SIGKDD*, pp 793–801
194. Li Y, Luo J, Chow C-Y, Chan K-L, Ding Y, Zhang F (2015) Growing the charging station network for electric vehicles with trajectory data analytics. In: *ICDE*, pp 1376–1387
195. Liu C, Deng K, Li C, Li J, Li Y, Luo J (2016) The optimal distribution of electric-vehicle chargers across a city. In: *ICDM*, pp 261–270
196. Guo L, Zhang D, Cong G, Wei W, Tan K-L (2016) Influence maximization in trajectory databases. *TKDE* 29(3):627–641
197. Liu D, Weng D, Li Y, Jie Bao Yu, Zheng HQ, Yingcai W (2016) Smartadp: visual analytics of large-scale taxi trajectories for selecting billboard locations. *TVCG* 23(1):1–10
198. Zhang P, Bao Z, Li Y, Li G, Zhang Y, Peng Z (2018) Trajectory-driven influential billboard placement. In: *SIGKDD*, pp 2748–2757
199. Zhang Y, Bao Z, Mo S, Li Y, Zhou Y (2019) Itaa: an intelligent trajectory-driven outdoor advertising deployment assistant. *VLDB* 12(12):1790–1793
200. Wang L, Zhiwen Yu, Yang D, Ma H, Sheng H (2019) Efficiently targeted billboard advertising using crowdsensing vehicle trajectory data. *TII* 16(2):1058–1066
201. Lou K, Yang Y, Wang E, Liu Z, Baker T, Bashir AK (2020) Reinforcement learning based advertising strategy using crowdsensing vehicular data. *TITS*
202. Wang M, Li H, Cui J, Deng K, Bhowmick SS, Dong Z (2016) Pinocchio: probabilistic influence-based location selection over moving objects. *TKDE* 28(11):3068–3082
203. Zhang D, Tao P, Karras Q, Li K, Jingbo Z, Hui X (2020) Geodemographic influence maximization. In: *SIGKDD*
204. Fangzhou S, Abhishek D, Jules W (2017) Dxnat-deep neural networks for explaining non-recurring traffic congestion. In: *Big Data*, pp 2141–2150
205. Lin Z, Fangce G, Rajesh K, John WP (2018) A deep learning approach for traffic incident detection in urban networks. In: *ITSC*, pp 1011–1016
206. Zhang Z, He Q, Gao J, Ni M (2018) A deep learning approach for detecting traffic accidents from social media data. *Transp Res Part C Emerg Technol* 86:580–596
207. Quanjun C, Xuan S, Harutoshi Y, Ryosuke S (2016) Learning deep representation from big and heterogeneous data for traffic accident inference. In: *AAAI*, pp 338–344
208. Yunjie L, Evan R, Joaquin C, Amir K, David L, Kenneth K, Michael W, William C, et al (2016) Application of deep convolutional neural networks for detecting extreme weather in climate datasets. *arXiv:1605.01156*
209. Sookyung K, Sasha A, Jiwoo L, Chengzhu Z, Aaron CW, Dean W (2017) Resolution reconstruction of climate data with pixel recursive model. In: *ICDMW*, pp 313–321
210. Evan R, Christopher B, Tegan M, Samira EK, Mr P, Chris P (2017) Extremeweather: a large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. In: *Advances in neural information processing systems*, pp 3402–3413
211. Shen M, Liu D-R, Shann S-H (2015) Outlier detection from vehicle trajectories to discover roaming events. *Inf Sci* 294:242–254

212. Wang Y, Qin K, Chen Y, Zhao P (2018) Detecting anomalous trajectories and behavior patterns using hierarchical clustering from taxi gps data. *GEIN* 7(1):25
213. Chao C, Daqing Z, Pablo SC, Nan L, Lin S, Shijian L (2011) Real-time detection of anomalous taxi trajectories from gps traces. In: *MobiQuitous*, pp 63–74
214. Jae-Gil L, Jiawei H, Xiaolei L (2008) Trajectory outlier detection: a partition-and-detect framework. In: *ICDE*, pp 140–149
215. Xin L, Changhu W, Jiang-Ming Y, Yanwei P, Lei Z (2010) Photo2trip: generating travel routes from geo-tagged photos for trip planning. In: *MM*, pp 143–152
216. Wei L-Y, Peng W-C, Lee W-C (2013) Exploring pattern-aware travel routes for trajectory search. *TIST* 4(3):1–25
217. Zaiben C, Heng TS, Xiaofang Z (2011) Discovering popular routes from trajectories. In: *ICDE*, pp 900–911
218. Wang S, Li M, Zhang Y, Bao Z, Tedjopurnomo DA, Qin X (2018) Trip planning by an integrated search paradigm. In: *SIGMOD*, pp 1673–1676
219. Chen C, Zhang D, Li N, Zhou Z-H (2014) B-planner: planning bidirectional night bus routes using large-scale taxi gps traces. *TITS* 15(4):1451–1465
220. Pinelli F, Nair R, Calabrese F, Berlingerio M, Di Lorenzo G, Sbodio ML (2016) Data-driven transit network design from mobile phone trajectories. *TITS* 17(6):1724–1733
221. Wang S, Zhifeng Bao J, Culpepper S, Sellis T, Cong G (2017) Reverse k nearest neighbor search over trajectories. *TKDE* 30(4):757–771
222. Bao J, He T, Ruan S, Li Y, Zheng Y (2017) Planning bike lanes based on sharing-bikes' trajectories. In: *SIGKDD*, pp 1377–1386
223. Zhou X, Chai C, Li G, Sun J (2020) Database meets artificial intelligence: a survey. *TKDE*
224. Vargas-Solar G, Zechinelli-Martini JL, Espinosa-Oviedo JA (2017) Big data management: What to keep from the past to face future challenges? *Data Sci Eng* 2(4):328–345
225. Alam M, Perumalla KS, Sanders P (2019) Novel Parallel Algorithms for Fast Multi-GPU-Based Generation of Massive Scale-Free Networks. *Data Sci Eng* 4:61–75
226. Li K, Li G (2018) Approximate Query Processing: What is New and Where to Go? *Data Sci Eng* 3:379–397
227. Huang W, Yu JX (2017) Investigating TSP Heuristics for Location-Based Services. *Data Sci Eng* 2:71–93
228. Gao D, Tong Y, She J et al (2017) Top-k Team Recommendation and Its Variants in Spatial Crowdsourcing. *Data Sci Eng* 2:136–150
229. Leal F, Malheiro B, González-Vélez H et al (2017) Trust-based Modelling of Multi-criteria Crowdsourced Data. *Data Sci Eng* 2:199–209
230. Dongo I, Cardinale Y, Chbeir R (2018) RDF-F: RDF Datatype inFerring Framework. *Data Sci Eng* 3:115–135
231. Lin P, Song Q, Wu Y (2018) Fact Checking in Knowledge Graphs with Ontological Subgraph Patterns. *Data Sci. Eng.* 3:341–358
232. Zheng Y, Wang J, Li G, Cheng R, Feng J (2015) QASCA: A quality-aware task assignment system for crowdsourcing applications. In *SIGMOD*, pp 1031–1046.
233. Fan J, Li G, Ooi BC, Tan K-L, Feng J (2015) icrowd: An adaptive crowdsourcing framework. In *SIGMOD*, pp 1015–1030
234. Li G, Wang J, Zheng Y, Franklin MJ (2016) Crowdsourced Data Management: A Survey. *IEEE Trans Knowl Data Eng* 28(9):2296–2319
235. Zheng Y, Li G, Cheng R (2016) Docs: Domain-aware crowdsourcing system. *PVLDB* 10(4):361–372
236. Zheng Y, Li G, Li Y, Shan C, Cheng R (2017) Truth inference in crowdsourcing: Is the problem solved? *PVLDB* 10(5):541–552
237. Li K, Zhang X, Li G (2018) A rating- ranking method for crowdsourced top-k computation. In *SIGMOD*, pp 975–990
238. Tian S, Mo S, Wang L, Peng Z (2020) Deep reinforcement learning-based approach to tackle topic-aware influence maximization. *Data Sci Eng* 5(1):1–11
239. Gharibshah Z, Zhu X, Hainline A, Conway M (2020) Deep learning for user interest and response prediction in online display advertising. *Data Sci Eng* 5(1):12–26
240. Wang Y, Yuan Y, Ma Y, Wang G (2019) Time-dependent graphs: Definitions, applications, and algorithms. *Data Sci Eng* 4(4):352–366
241. Wang Y, Yao Y, Tong H, Xu F, Lu J (2019) A brief review of network embedding. *Big Data Min Analytics* 2(1):35
242. Li J, Li M, Wang H (2020) Mining conditional functional dependency rules on big data. *Big Data Min Analytics* 03(1):68
243. Qin X, Luo Y, Tang N, Li G (2018) Deepeye: An automatic big data visualization framework. *Big Data Min Analytics* 1(1):75
244. Yuan H, Li G, Feng L, Sun J, Han Y (2020) Automatic view generation with deep learning and reinforcement learning. In *ICDE*, pp 1501–1512.
245. Nathan V, Ding J, Alizadeh M, Kraska T (2020) Learning multi-dimensional indexes. In: *SIGMOD*, pp 985–1000
246. Li P, Lu H, Zheng Q, Yang L, Pan G (2020) Lisa: a learned index structure for spatial data. In: *SIGMOD*, pp 2119–2133
247. Yuan H, Li G, Bao Z, Feng L (2021) An effective joint prediction model for travel demands and traffic flows. In: *ICDE*
248. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Jill B, Christy D, Thamar S (eds), *NAACL-HLT*, pp 4171–4186
249. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler DM, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020) Language models are few-shot learners. In: Hugo L, Marc'Aurelio R, Raia H, Maria-Florina B, Hsuan-Tien Li (eds) *NeurIPS*
250. Jing Yuan Yu, Zheng XX, Sun G (2013) T-drive: enhancing driving directions with taxi drivers' intelligence. *TKDE* 25(1):220–232