A Survey on Modern Deep Neural Network for Traffic Prediction: Trends, Methods and Challenges

David Alexander Tedjopurnomo, Zhifeng Bao, Baihua Zheng, Farhana Murtaza Choudhury, and A. K. Qin

Abstract—In this modern era, traffic congestion has become a major source of severe negative economic and environmental impact for urban areas worldwide. One of the most efficient ways to mitigate traffic congestion is through future traffic prediction. The research field of traffic prediction has evolved greatly ever since its inception in the late 70s. Earlier studies mainly use classical statistical models such as ARIMA and its variants. Recently, researchers have started to focus on machine learning models because of their power and flexibility. As theoretical and technological advances emerge, we enter the era of deep neural network, which gained popularity due to its sheer prediction power which can be attributed to the complex and deep structure. Despite the popularity of deep neural network models in the field of traffic prediction, literature surveys of such methods are rare. In this work, we present an up-to-date survey of deep neural network for traffic prediction. We will provide a detailed explanation of popular deep neural network architectures commonly used in the traffic flow prediction literatures, categorize and describe the literatures themselves, present an overview of the commonalities and differences among different works, and finally provide a discussion regarding the challenges and future directions for this field.

Index Terms—Deep neural network, deep learning, traffic flow prediction, traffic speed prediction, road network

1 Introduction

Traffic congestion is a major problem faced by metropolitan cities. In 2015, it is estimated that the avoidable cost of traffic congestion for Australian capital cities is approximately $16.5 billion, up from the 2010 estimate of $12.8 billion. Furthermore, this value is estimated to increase to about $30 billion by 2030 [1]. Most congestion mitigation measures are costly, difficult to implement, or both. For instance, Singapore implemented regulations on the number of vehicles on roads [2], which is infeasible for countries with poor public transportation systems. Constructing new roads to ease congestion is also difficult due to the extremely high cost. As an example, the estimated per mile cost of a standard one lane road in New Jersey, USA is $220,490 [3].

With the advancements and widespread adoption of traffic sensors, access to large traffic databases is now available. This has led to the development of traffic prediction as a research field. Educated traffic decision made through accurate prediction is a far cheaper and easier to implement alternative for reducing road congestion. Future traffic prediction involves creating a prediction model from historical traffic data to predict the short-term future traffic state ranging from 5 to 60 minutes into the future. Traffic prediction is different from conventional time-series analysis in that traffic prediction is subject to spatial factors as well as many other external factors. For instance, the prediction of traffic at one site depends on the traffic at other sites and all of the sites are affected by external factors such as weather and holidays.

Amongst all the available traffic prediction methods, deep neural network is the most prominent. This is due to its sheer predictive power that can model the complex and nonlinear traffic patterns [4], [5], [6], [7], [8]. The three most common deep neural network models used for traffic prediction are Convolutional Neural Networks, Recurrent Neural Networks, and Feedforward Neural Networks. The increasing popularity of deep neural network models for traffic prediction has led to numerous publications, but issues such as the wide variety of hybrid deep neural network structures have made it difficult to assess the current state and future directions of this research field. This problem is compounded by the fact that survey works focusing specifically on deep neural network models are rare. In this work, we attempt to address these issues by presenting a comprehensive overview of the area. The main audience for our paper are practitioners interested in applying deep neural networks to the problem of traffic prediction. As such, we have organized our paper accordingly. We will first outline the problem definition and a short history of traffic prediction. Then, we will describe the three most popular deep

- D. A. Tedjopurnomo and Z. Bao are with the RMIT University, Melbourne, Victoria 3000, Australia.
E-mail: {david.tedjopurnomo, zhifeng.bao}@rmit.edu.au.
- B. Zheng is with Singapore Management Univerity, 188065, Singapore. E-mail: bhzheng@smu.edu.sg.
- F. M. Choudhury is with the University of Melbourne, Parkville, Victoria 3010, Australia. E-mail: farhana.choudhury@unimelb.edu.au.
- A. K. Qin is with the Swinburne University of Technology, Hawthorn, Victoria 3122, Australia. E-mail: kqin@swin.edu.au.

*Manuscript received 25 May 2019; revised 4 Mar. 2020; accepted 5 June 2020. Date of publication 9 June 2020; date of current version 7 Mar. 2022.
(Corresponding Author: Zhifeng Bao.)
Recommended for acceptance by L. Chen.
Digital Object Identifier no. 10.1109/TKDE.2020.3001195*

neural network models used in traffic prediction research. Afterwards, we will discuss traffic prediction by listing out, categorizing and discussing 37 state-of-the-art deep neural network for traffic prediction literatures based on the dataset and the model. These literatures mainly cover work from the transportation research field such as the Transportation Research Part C journal, but we also cover several publications from the neural network and knowledge management field. All of the covered literatures are from the years 2014 to 2019. Finally, we will discuss the present and future challenges facing this research field. The insights that readers can extract from this paper are:

- The deficiencies of current traffic prediction survey work, especially with regards to the models covered.
- The development of the traffic prediction research field from its beginnings in the 1970s.
- The strengths and weaknesses of the three most common deep neural network models used in traffic prediction.
- The commonly used datasets in traffic prediction research, the associated parameters and how these affect the prediction task.
- The different ways current literatures utilize the deep neural network models for the prediction task.
- The current challenges facing the traffic prediction task and how these challenges have been solved, or partially solved, by the introduction of deep neural network methods.
- The future challenges facing traffic prediction and how to deal with these challenges.

In Section 2, we first provide an overview and history of the traffic prediction field. Section 3 describes the workings of the three main deep neural network models: Convolutional Neural Networks, Recurrent Neural Networks and Feedforward Neural Networks. Our work is focused on Section 4, where we categorize and discuss the literature. We split this section into two: Section 4.1 discusses the datasets used in the literature, while Section 4.2 discusses the models. We also provide a short discussion on Section 4.3. Then, in Section 5, we will describe the challenges of traffic prediction research. Section 5.1 describes the current challenges of traffic prediction research and how they have been addressed or partially addressed through the adoption of deep neural networks. Afterwards, Section 5.2 discusses the future challenges and how they can be addressed. Finally, we conclude our work in Section 6.

Comparison to Other Survey Work. One of the most important literature surveys of this field is the work of Vlahogianni et al. [9]. Their work mainly discussed the challenges of traffic prediction, focusing more on the research field rather than the models. Additionally, the authors covered the literatures from 2006 to 2013, which do not include the now ubiquitous deep neural network models. Another difference is in the model taxonomy; their work categorized the models based on several criteria such as the type of model (e.g., statistical, neural network, hybrid model) and the problem (e.g., time series, function approximation). This taxonomy is outdated because modern traffic prediction models are mainly based on deep neural network, which under their taxonomy will all fall under the neural network category of model and function approximation category of problem. Our taxonomy on the other hand, is designed to provide a more up-to-date categorization of models.

A recent paper by Nagy and Simon [10] is a more up-to-date survey on traffic prediction. They provided an overview of the different types of models used for this task. However, their model taxonomy only has a few points of comparison, which are: whether or not the model integrates environmental data, contains spatial property, handles nonlinearity and handles nonstationarity. We perform a more comprehensive comparison on both the models and the data, totaling eleven points of comparison combined. Additionally, their work does not have a future challenges section that discusses how the field can be advanced. We provide this discussion in Section 5.

The work of Zhu *et al.* [11] provides another up-to-date survey of the field. However, their work focuses on big data analytics without much focus on the actual models. Our work provides a more balanced approach by discussing both the models and the datasets in Section 4. We also discuss the field as a whole, through the discussion of future challenges, in Section 5.

Finally, we would like to express the importance of comparisons between different hybrid deep neural network model implementations. Due to the availability of deep neural network libraries such as Keras [12], PyTorch [13], and TensorFlow [14], development of complex neural network models has become much easier. Consequently, the trend is to use different hybrid models to capture the different aspects of the data, such as the temporal aspect and the spatial aspect. Because of this reason, it is very important to perform a thorough comparison among different hybrid models that capture different aspects of the data, or even the same aspects using different ways. To the best of our knowledge, our work is the first one to attempt such task.

2 Background

In this section, we first describe the problem formulation of traffic prediction. Then, we briefly outline the history of traffic prediction and show why deep neural network became the benchmark category of methods.

Traffic prediction concerns the usage of a learnable function that takes as input the historical traffic data from several previous time-steps in order to predict the traffic in the future. Two main types of traffic data used are traffic flow and traffic speed. Traffic flow is denoted as the total number of vehicles detected in a target detection site during a certain time period. Traffic speed is denoted as the average traveling speed of vehicles detected in a target detection site during a certain time period. In this section, we will use the general term "traffic" to refer to both traffic flow and traffic speed. The traffic prediction problem can be denoted as:

$$\hat{y}_{t+T'} = f([X_{t-T-1}, X_{t-T}, \ldots, X_t])$$

The objective is to find the model parameters which minimize the error between the predicted traffic and the observed traffic:

$$\theta^* = \arg\min_{\theta^*} L(y_{t+T'}, \hat{y}_{t+T'}; \theta^*)$$

- y_t : The observed traffic at time t
- $\hat{y}_t$: The predicted traffic at time t
- T : Input sequence length, i.e., how many time steps of past traffic data are used as the input.
- T' : Prediction horizon, i.e., how many time steps in the future the prediction is for.
- f : An arbitrary function that calculates the traffic prediction based on the input data.
- L : Loss function, which is the function that calculates the quality of the prediction.
- θ^* : The optimal set of parameters for the function f

All of f, L, and θ^* depend on the actual model used. We will now discuss the different types of prediction models that have been used for traffic prediction in the past.

The field of traffic prediction has existed for almost five decades and covers a wide array of methodologies which can be divided into three main categories. The first category belongs to the classical statistical models, of which the Autoregressive Integrated Moving Average (ARIMA) family of models is the most popular. Ahmed and Cook are the first researchers to apply ARIMA to traffic prediction [15]. Shortly after, Levin and Tsao [16] applied ARIMA on two freeway locations and found that the ARIMA(0,1,1) model is the most statistically significant.

Other authors also applied different versions and improvements to ARIMA. Lee and Fambro [17] applied subset ARIMA and found that it provides stable and accurate results. Williams [18] discovered the impact of upstream traffic sensors to downstream ones and applied ARIMAX model for traffic flow prediction. Williams and A. Hoel [19] applied Seasonal ARIMA to the United States and the United Kingdom traffic data. Kamarianakis and Prastacos [20] discussed and compared the Vector Autoregressive Moving Average and Single Space-Time ARIMA model.

Despite the popularity, classical statistical models are relatively weak. This is because they are simple linear models which assume that the traffic is stationary. Consequently, they frequently fail when handling the complex, nonlinear traffic data [6], [8], [21], [22]. Additionally, these models were proposed at a time where traffic data were simpler and much smaller in size [23], a condition that no longer holds true in the present day where the ubiquity of traffic sensors has caused an explosion in traffic flow data.

Due to the aforementioned deficiencies of classical statistical models, researchers flocked to machine learning models. Machine learning models are flexible as they can learn from the data. That is, the parameters of the prediction function are adjusted automatically as the model traverses through the dataset, as opposed to the classical statistical models in which the function parameters are manually defined a priori [24]. The main weakness is that machine learning models are data intensive [25]. However, as previously mentioned, large traffic flow data are now available. For more differences between classical statistical models and machine learning models, we refer readers to the work of Karlaftis and Vlahogianni [21].

Out of the different machine learning models, neural network is the most commonly used. The reason behind its prominence is that many other machine learning models' feature extraction phase, which helps extract useful patterns and information from the data to help the prediction, is done manually (i.e., using manually tuned kernels). On the other hand, neural networks perform automatic feature extraction as well as the actual prediction in one model.

One of the first neural network applications in traffic flow prediction was by Dougherty *et al.* [26]. Since then, various improvements to the neural network structure have been proposed. Vlahogianni *et al.* [27] proposed a genetic algorithm approach to optimally tune the network. Zheng and Lee [28] used multiple neural network predictors which are combined using the theory of conditional probability and the Bayes rule. Time delay neural network model was applied to traffic prediction in 2005 by Zhong *et al.* [29]. Chan *et al.* [30] imbued a neural network model with the hybrid exponential smoothing method to preprocess training data and the Levenberg-Marquardt algorithm to train the network weights. Other types of machine learning models aside from neural network were also used, such as the k-Nearest Neighbor [31], [32], [33] and the Support Vector Regression [34], [35], [36].

While machine learning models, and especially neural network, are more powerful compared to statistical models, they are very hard to train efficiently. Thus, machine learning models during the 2000s utilize shallow and simple structures, limiting their prediction power. However, the increasing computational power, as well as theoretical and software improvements in recent times had made increasingly complex neural network models feasible to train. Thus, in the middle of the 2010s, researchers started to apply deep neural network models for traffic prediction.

Deep neural networks consist of complex neural network models with a large number of layers. Some examples are Recurrent Neural Network, Convolutional Neural Network, Feedforward Neural Network, and hybrids of these models. Some of the deep neural network models can explicitly capture different aspects of traffic data, which made them even more attractive. For instance, CNN can explicitly capture the spatial aspect of traffic data while RNN can explicitly capture the temporal aspect of traffic data. Additionally, the increased number of layers improves the models' prediction capability. This factor allows them to model traffic fluctuations more accurately.

While the strengths of deep neural network models made them attractive, they also possess several disadvantages compared to the older methods:

- *Deep Neural Network Models Require a Large Amount of Data That Covers all Traffic Conditions.* If the amount of data is too small or if the data is not diverse enough, the model's generalization capability is compromised.
- *Deep Neural Network Models Still take a Long Time to Train.* As deep neural network models are complex and have a large number of layers, the training time can be very long. This problem is compounded on hybrid deep neural network models. As classical statistical and older machine learning models are not as complex, their training time is much shorter.
- *Deep Neural Network Models are Difficult to Interpret.* This is because of two reasons: the number of internal parameters is very large, and the parameters are learned from training, not set manually. Thus, while

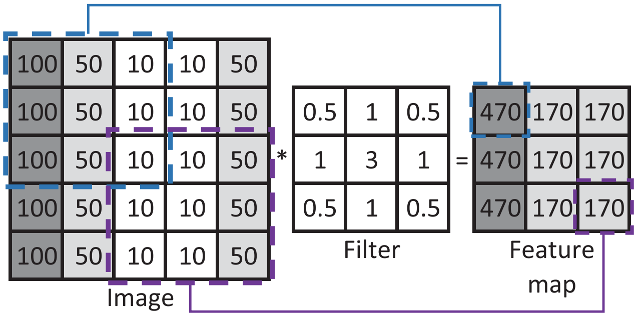


Fig. 1. An example of a convolution process.

they can predict well, it is hard to understand their parameters. Understanding the parameters may reveal important information such as the spatiotemporal dynamics in the road network.

Summary. Traffic prediction is a task of training an arbitrary function to predict future traffic given past traffic data. The earliest class of models used is the classical statistical models. Afterwards, machine learning models improve upon the performance of classical statistical models. Then, the deep neural network class of models dominates the field due to its capability in capturing the complex and nonlinear patterns in traffic data.

3 Deep Neural Network

In the following subsections, we will discuss different core deep neural network structures, their intuitions, and applications in the context of traffic prediction.

3.1 Convolutional Neural Network

A Convolutional Neural Network has the capability to learn inherent features progressively, starting from low level features and then building up to more abstract concepts through a series of convolutional layers. Although this strength contributes to its popularity in image recognition, CNNs have been regularly applied to traffic flow prediction. The intuition is that traffic flow readings can be modeled as an image, where each pixel corresponds to the traffic intensity at a certain block of area. Thus, similar techniques developed for image recognition can be easily applied.

A CNN consists of several "convolution" and "pooling" layers. Convolution's purpose is to extract features from the input, whereas pooling's purpose is to reduce the dimensionality of each feature map but preserve the most important information. Given a road network, the input of a CNN is preprocessed by partitioning the network as a grid, which is essentially a set of cells with each cell representing an area in the data space and the value associated with the cell representing the number of vehicles detected in that cell at a certain point in a time period (e.g., 5×5 cells in Fig. 1). The traffic flow reading for each time period will be represented with the same grid but different number of vehicles. Thus, the entire traffic data modeled this way can be seen as several images with the same size but different pixel values.

Applying the convolution and pooling layers results in a smaller output that represents higher-level latent features. As an example, in traffic flow prediction, the first few layers may summarize the traffic condition of several city blocks. Further applications may summarize the traffic of these city blocks into traffic condition for an entire city district and so on. Mathematically, convolution layers extract features by computing the dot product between a matrix of some preset values (referred to as filter) and a subset of cells from the original grid, which produces a matrix that is called feature map. The example in Fig. 1 shows, (i) the top-left 3×3 subset of cells produces the value 470, and (ii) the bottom-right 3×3 subset of cells produces the value 170 in the feature map. This can be interpreted as the top-left subset having a much higher number of vehicles in that region than the bottom-right subset.

Unlike most neural networks, CNN's layers are not fully connected. Consequently, the number of parameters and training time are significantly reduced. Additionally, CNN uses a weight sharing mechanism, which further reduces the number of required parameters. Since CNN's layers are not fully connected, one layer of CNN does not learn from all of the previous layer's features. However, this actually proves to be an advantage in many applications as CNN can learn how the different parts of the input relate to each other spatially.

In the application of traffic prediction, CNN is often used as a component in a hybrid deep neural network, whose task is to capture the spatial aspect of traffic data. This is because different roads in different locations may be correlated and these correlated roads share similar traffic trend. Therefore, the traffic of the correlated roads may rise or fall, depending on their historical data [37]. For instance, during the evening, there is a strong correlation between the road traffic of commercial and residential districts because employees are heading off from work.

3.2 Recurrent Neural Network and Long Short-Term Memory

Recurrent Neural Networks are commonly applied to sequence data because of their memorization capability, which can learn both long and short term dependencies between parts of the sequence. Additionally, RNN is able to scale to longer sequences compared to other network architectures. Its unique capability makes it one of the most popular deep neural networks.

An RNN consists of a single node with a recurrent connection, but is often visualized as a chain of nodes, with each node representing the network state at a particular recurrence/time step. This visualization can be seen in Fig. 2. The node state s_t processes the input data x_t at time t, as well as a 'summary' of all the information obtained up to time $t-1$. This summary is stored in s_{t-1}, and it memorizes which parts of the sequence are important. Node s_t then has the summary up to time t and this information is passed to the next node state s_{t+1}. Thus, the node state s_t stores the state of nodes for all the previous time steps until the beginning of the input (i.e., s_{t-1}, s_{t-2}, ...). The output o_t is then compared with the ground truth y_t in order to calculate the loss, which is used to fine-tune the model parameters. In traffic prediction applications, the input to an RNN consists of past traffic readings. A continuous time period is divided into discrete time blocks and the traffic flow reading from each block is fed into the RNN.

By its nature of being able to take in possibly very long sequences, RNN suffers from the vanishing gradient problem,

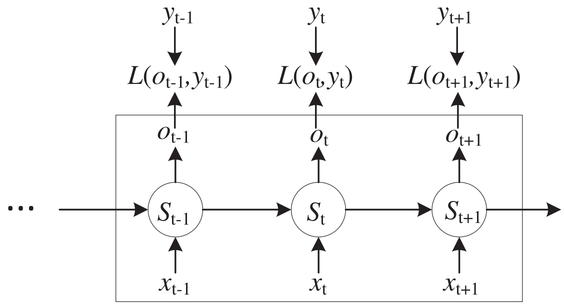


Fig. 2. A recurrent neural network.

which hinders the network's ability to memorize information for a long time. For this reason, Hochreiter and Schmidhuber [38] proposed the Long Short-Term Memory (LSTM), which was further improved in [39].

LSTM also contains multiple layers, each possessing a cell with the memorization capability. In addition, it contains three gates, which control how information propagates throughout the network. These gates are the input gate i, which controls the importance of the inputs x_t and h_{t-1}, forget gate f which controls how much of the previous information C_{t-1} is to be forgotten, and the output gate, which controls how relevant is the current information C_t for the next step. As can be seen in Fig. 3, it maintains the RNN's recurrent structure, but introduces the three gates to control the cell value.

RNN-based methods in general possess the major advantage in the form of its memorization capability. The ability to learn important parts of the sequence and then knowing when to memorize or forget them had led RNN to be the prime choice for sequence data. Due to this, RNN based models have been applied in many fields such as named entity recognition [40], voice recognition [41], music composition [42], and image caption generation [43]. However, RNN's recurrent structure leads to significantly longer training time compared to other deep neural network models.

In the field of traffic prediction, LSTM as well as other variants of RNN-based methods are commonly used as a component in hybrid deep neural network models. Its task is to capture the temporal patterns of traffic data; learning how traffic evolves over time.

3.3 Feedforward Neural Network

A Feedforward Neural Network, which is also commonly referred to as Fully Connected Neural Network (FC or FCNN), is one of the earliest and simplest neural network models. It consists of several layers of fully connected computational nodes organized in many layers. The value of every node in the hidden or output layers is computed by taking the weighted sum of all of the previous layer's nodes and then passing the value to a nonlinear function such as sigmoid, tanh and relu.

The FNN's fully connected structure enables each of its layers to process the combination of all the previous layer's features. However, this also serves as a weakness because its full connection results in a large amount of parameters. Consequently, the training process of FNNs can be quite time consuming. In addition, FNNs do not have the capability of explicitly capturing spatial or temporal data. Because of this, FNNs are rarely used as the main predictor in deep neural network literatures.

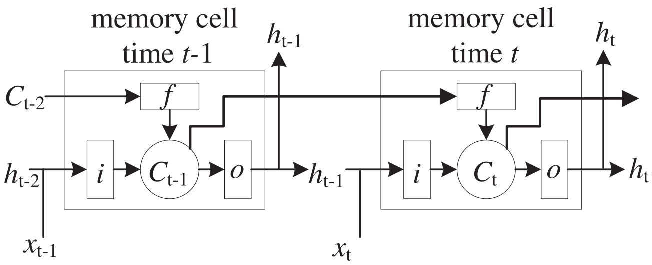


Fig. 3. Visualization of how information propagates through an LSTM.

For traffic flow prediction, FNNs usually serve a utility role in a hybrid deep network, whose main purpose is to perform tasks such as aggregating outputs from different components within the network, dimensionality transformation and incorporating external data such as weather. This is because the size of input layer or output layer can be set manually, which gives FNN the capability to transform inputs of an arbitrary dimensionality to an output of an arbitrary dimensionality. When used to integrate external data, the input depends on the type of external data. Numerical values can be provided as it is while categorical values need to be transformed first (e.g., using one-hot encoding). For aggregating outputs and dimensionality transformation, the inputs depend entirely on the model. More details on FNN's application in the traffic prediction domain can be found in Section 4.2.3.

Summary. In this section, we described three popular deep neural network architectures, their strengths, weaknesses, and applications. RNN is commonly used to capture the temporal trends of traffic data–the dynamics of how past traffic can influence future traffic. CNN is commonly used to capture the spatial trends of the data–how traffic propagates through the road network. FNN can aggregate the output from different subnetworks and also can process external data such as weather information. We will describe the typical usage of these models in Section 4.2.

4 Deep Neural Network for Traffic Flow Prediction

In this Section, we will describe 37 literatures and the methodologies used to predict traffic flow. We only consider recent (2014 to 2019) papers that provide sufficiently novel improvements and contributions to the field. In the first subsection, we will discuss the datasets in terms of the main and secondary datasets, as well as the dataset-related parameters and how they affect the prediction task. Then, in the second subsection, we discuss how these different deep neural network models are used.

4.1 Traffic Flow Prediction – Data

Tables 4 and 5 provide an overview of 37 existing works, with each column representing a decision that researchers have to make with regards to traffic flow prediction data. Do note that when a work uses more than one datasets, the settings for those datasets may differ. Hence, we use numbers to denote the different datasets and their settings. For instance, [44] uses the LA traffic data from Mar 2012 to Jun

TABLE 1
Popular Main Datasets

Main Dataset	Data Type	Sample References	Popularity
Caltrans PEMS	Point	[5], [50], [51], [52]	14 out of 37
Beijing dataset	Point	[45], [46], [47]	6 out of 37
Beijing dataset	Trajectory	[4], [53], [54]	6 out of 37

TABLE 2
Popular Secondary Datasets

Secondary Dataset	Sample References	Popularity
Weather	[47], [52], [56]	6 out of 37
Time of day/day of week	[55], [56]	3 out of 37
Road network	[7], [57]	3 out of 37

2016, and the Caltrans PEMS data from Jan 2017 to May 2017. Thus, we assign the number 1 to LA traffic data and number 2 to Caltrans PEMS data, and their corresponding settings in other applicable columns will be represented with these numbers as well. In the case of [5], they use two different datasets, but both have the same parameters, so we only use the numbers for the "Primary Dataset" column.

Main Dataset. As can be observed from Tables 4 and 5, the Caltrans data[1] is by far the most commonly used dataset. This is because of its public availability, ease of download, simple structure and long historical data. The Caltrans data provides information regarding date, time stamp, traffic flow per lane, and aggregated traffic flow. Traffic flow is the most commonly used field, but occupancy and speed information is also available. The data granularity can be set to 5 minutes, hourly, daily, weekly and monthly depending on user requirements.

Numerous authors use data from Beijing. However, it is unclear as to whether or not all of the Beijing-based datasets come from one unified dataset source. These datasets usually cover the Ring Road area, as can be seen in the works [45], [46], [47], [48] and [49]. The dataset used by Ma *et al.* [45] contains the traffic volume, occupancy and speed data, similar to Caltrans.

Unlike point data that has a dataset being considered as the standard, the trajectory data used in these traffic prediction experiments do not have a standard dataset. Different works use different datasets with different properties, including the origin country (mostly America or China), method of transportation (cars, taxis or bicycles) and time range. Amongst these works, trajectory data from Beijing is relatively more popular. We summarize the top-3 most popular main datasets in Table 1, which cover 70 percent of the literature works we surveyed. Please take note that many works use more than one main datasets.

Secondary Dataset. Secondary dataset is not commonly used in the literate studies. Among the 37 works we surveyed, only 10 use secondary dataset, as observed from Tables 4 and 5. We list the top-3 popular secondary datasets in Table 2. Please take note that multiple secondary datasets might be used by one work.

TABLE 3
Data Time Ranges

Data Time Range	Sample References	Popularity
One month	[45], [53]	5 out of 37
Several months	[50], [52]	22 out of 37
One year	[5], [58]	6 out of 7
More than one year	[6], [54]	4 out of 37

According to our observations, the low usage of secondary dataset is mainly caused by the difficulty of integrating the main data with the secondary data. For instance, one model that uses the Caltrans data covering a long highway will need to match the time stamp, the latitude, and the longitude of each reading in order to find the appropriate weather and accidents data. This task is very difficult. Furthermore, there is some added time complexity of aggregating the different data together, which is undesirable, especially in an already time-consuming hybrid deep neural network structure.

Conversely, time-of-day and day-of-week data are much easier to incorporate. They can be very useful as the inclusion of time-of-day data allows the model to learn the difference between traffic conditions during various periods within a day while the inclusion of day-of-week data allows the model to learn the traffic patterns of different days, which is especially important in distinguishing weekdays and weekends traffic. These features are used by Yu *et al.* [55], and their experiments show that the inclusion of these factors improves the prediction performance.

Data Time Range. One major deficiency we have observed in the field of deep neural network for traffic flow prediction is the insufficient data time range. 26 out of 37 literatures in Tables 4 and 5 use less than one year's worth of data. This deficiency will have an adverse impact on subtropical regions, as seasonal changes may affect temperature and weather, which in turn can affect traffic. By using data from only one or a few months, the model cannot generalize to different seasons. This can be mitigated by incorporating weather data, but as mentioned before, this is a difficult and time-consuming task.

Some authors also use data from only a certain range of hours or use data from weekdays only. This will also cause problems as the model cannot generalize well to situations outside the boundaries of the provided data. For instance, using traffic data from 07.00 AM to 11.00 PM only may reduce the model's performance on the excluded hours, and using only weekdays data may adversely impact the model's performance when predicting weekend traffic. In Table 3, we summarize the data time ranges from the chosen literatures.

Deep neural network models are flexible and can adapt well to data. Consequently, using vastly different datasets may result in an entirely different model. Thereby, for a model to be applicable to real application scenarios, it is important to use a dataset that closely resembles those scenarios.

Data Granularity. Most of the literatures in Tables 4 and 5 use a data granularity of 5 minutes. This is likely caused by the availability of that granularity as a default setting in common datasets such as Caltrans. Although, The

[1] http://pems.dot.ca.gov/

TABLE 4
Data Categorization for the Covered Literature

Reference	Authors	Year	Primary Datatype	Primary Dataset	Data Time Range	Data Granularity	Secondary Dataset	Input Sequence Length	Prediction Horizon
[5]	Huang et al.	2014	Point	1) Caltrans PEMS 2) China highways	2011	15 minutes	None	60 minutes	15 minutes
[50]	Yisheng et al.	2015	Point	Caltrans PEMS	Jan - Mar 2013 (weekdays)	5 minutes	None	a) 15 minutes b) 15 minutes c) 20 minutes d) 15 minutes	a) 15 minutes b) 30 minutes c) 45 minutes d) 60 minutes
[45]	Ma et al.	2015	Point	Beijing Ring Road	June 2013	2 minutes	None	2 minutes	2 minutes
[51]	Tian and Pan	2015	Point	Caltrans PEMS	2014 (workdays)	5 minutes	None	a) 180 minutes b) 180 minutes c) 180 minutes d) 180 minutes	a) 15 minutes b) 30 minutes c) 45 minutes d) 60 minutes
[46]	Jia et al.	2016	Point	Beijing traffic	Jun - Aug 2013	2 minutes	None	a) 16 minutes b) 24 minutes c) 50 minutes	a) 2 minutes b) 10 minutes c) 30 minutes
[52]	Soua et al.	2016	Point	Caltrans PEMS	1 Aug 2013 - 25 Nov 2013	15 minutes	- Weather data - Tweets data	Unknown	Unknown
[53]	Wang et al.	2016	Trajectory	Beijing taxi	November 2013 (weekdays)	5 minutes	None	10, 20, 30, 40 and 50 minutes	10, 20, 30, 40 and 50 minutes
[6]	Wu and Tan	2016	Point	Caltrans PEMS	Apr 2014 - Jun 2015	5 minutes	None	75 minutes (short-term) 30 minutes (daily/weekly)	5 minutes
[7]	Cheng et al.	2017	Traffic condition	Beijing map app	Mar - Jun 2016	5 minutes	- Road network data - Speed limit data	a) 15 minutes b) 30 minutes c) 60 minutes d) 90 minutes	a) 15 minutes b) 30 minutes c) 45 minutes d) 60 minutes
[8]	Dai et al.	2017	Point	Caltrans PEMS	First 16 weeks of 2016	5 minutes	None	60 minutes	5 minutes
[59]	Du et al.	2017	Point	Caltrans PEMS	1 Feb 2013 - 29 Aug 2013	5 minutes	None	100 minutes	5 minutes
[60]	Fouladgar et al.	2017	Point	Caltrans PEMS	15 Aug 2016 - 14 Oct 2016	30 minutes	None	150 minutes	30 minutes
[47]	Jia et al.	2017	Point	Beijing traffic	June - August 2013	2 minutes	Weather data	DBN: a) 10 minutes b) 18 minutes c) 22 minutes LSTM: a) 12 minutes b) 20 minutes c) 24 mintues	a) 2 minutes b) 10 minutes c) 30 minutes
[61]	Kang et al.	2017	Point	Caltrans PEMS	Oct - Nov 2009	5 minutes	None	Unknown	15, 30, 60 minutes
[44]	Li et al.	2017	Point	1) LA traffic 2) Caltrans PEMS	1) Mar 2012 - Jun 2016 2) Jan 2017 - May 2017	5 minutes	None	Unknown	15, 30 and 60 minutes
[54]	Ma et al.	2017	Trajectory	Beijing Taxi GPS	1 May 2015 - 6 Jun 2015	2 minutes	None	a) 30 minutes b) 40 minutes c) 30 minutes d) 40 minutes	a) 10 minutes b) 10 minutes c) 20 minutes d) 20 minutes
[55]	Yu et al.	2017	Point	Caltrans PEMS	19 May 2012 - 30 Jun 2012	5 minutes	- California accidents - Los Angeles accidents - Time of day - Day of week	1 week	a) 5, 15, 30, 60 minutes (w/o accident data) b) 5, 30, 60, 90, 120, 150, 300 minutes (w/ accident data)

Highways Capacity Manual [75] recommended a data granularity of 15-minutes, which saw some authors aggregate the 5-minute readings from Caltrans to 15-minute readings.

Depending on the dataset, the data granularity is a potentially important hyperparameter. Using a data granularity that is too small may cause a lot of zero values, especially during conditions where traffic is very sparse. For example, it is highly likely for a traffic loop detector to not detect any cars in 2 or 5 minute periods during off-peak hours (e.g., 02:00-04:00 AM) while this becomes less likely if the granularity is increased to 15 minutes or more. On the other hand, using a granularity that is too high might result in the smoothness of the traffic flow reading where important trends are lost. For instance, if the traffic experiences periodic shifts during 12:30PM, this trend might not be detected if the data granularity is one hour.

Data granularity also impacts the number of possible data points as well as the size of the input sequence. Using a smaller granularity will increase the length of the required data sequence. For instance, one hour's worth of data can be captured with only a sequence of length 4 when the granularity is 15 minutes, but when the granularity is 5 minutes, the sequence length is 12. This can impact training time, especially for RNN-based models.

Due to the aforementioned reasons, choosing the correct data granularity becomes a decision based on trade-offs and should be considered carefully depending on the data, the model, as well as the application scenarios.

Input Sequence Length and Prediction Horizon. As can be seen in Tables 4 and 5, many authors perform experiments with different prediction horizons and use different input sequence lengths for each of the selected prediction horizons. Hence, for the works that use a certain input sequence length for a certain prediction horizon, we use the alphabets to denote that these parameters are paired. For instance, in [50], for predicting traffic 15 minutes in the future, they use 15 minutes of input data and for predicting the traffic 45 minutes into the future, they use 20 minutes of input data. There are some works such as [53] where the alphabets are not used. In this case, the input sequence length and prediction horizon are parameters that the authors explore separately.

Intuitively, as we increase the prediction horizon, the input sequence length also needs to be increased. This is because the increase in prediction horizon means predicting the traffic of further time frame in the future and thus, increasing the task complexity. Increasing the size of the

TABLE 5
Data Categorization for the Covered Literature *Contd.*

Reference	Authors	Year	Primary Datatype	Primary Dataset	Data Time Range	Data Granularity	Secondary Dataset	Input Sequence Length	Prediction Horizon
[4]	Yu et al.	2017	Trajectory	Beijing floating cars	Jun - Aug 2015 (6 AM - 10 PM)	2 minutes	None	30 minutes	- 2, 4, 6 minutes (short term) - 20, 40, 60 minutes (long term)
[48]	Yu et al.	2017	Point	1) Beijing loop detectors 2) Caltrans PEMS	1) Jul - Aug 2014 2) May - Jun 2012 (weekdays)	5 minutes	None	60 minutes	15, 30 and 45 minutes
[56]	Zhang and Kabuka	2017	Point	Caltrans PEMS	Nov - Dec 2016	60 minutes	Weather	100 hours	12 hours
[62]	Zhang et al.	2017	Trajectory	1) Beijing taxi 2) New York bike	1) Jul - Nov 2013, Mar - Jun 2014, Mar - Jun 2015, Nov 2015 - Apr 2016 2) Apr - Sep 2014	1) 30 minutes 2) 1 hour	- Weather - Day of week	Short: 90, 120, 150 minutes Medium: Previous 1, 2, 3, 4 days Long:Previous 1, 2, 3, 4 weeks	30 minutes and 1 hour
[49]	Zhao et al.	2017	Point	Beijing traffic	Jan 2015 - Jun 2015	5 minutes	None	a) 10 minute b) 15 minutes c) 25 minutes d) 30 minutes	a) 15 minutes b) 30 minutes c) 45 minutes d) 60 minutes
[63]	Cui et al.	2018	1) Point 2) Road link	1) Seattle traffic 2) INRIX GPS	1) 2015 2) 2012	5 minutes	None	50 minutes	5 minutes
[23]	Cui et al.	2018	1) Point 2) Road link	1) Seattle traffic 2) INRIX GPS	2015	5 minutes	None	50 minutes	5 minutes
[58]	Kim et al.	2018	Point	Santander city traffic	2016	15 minutes	None	15 minutes	150 and 210 minutes
[57]	Liao et al.	2018	Point	Beijing traffic	April - May 2017	15 minutes	- Map query data - Events data - Road network	1 day	2 hours
[64]	Ren et al.	2018	Point	Singapore traffic	182 days (date not mentioned)	5 minutes	None	60 Minutes	5, 10, 15 and 20 minutes
[65]	Wang et al.	2018	Trajectory	1) Washington D.C. bike 2) Chicago bike	2015 - 2016	30 minutes	- Check-in data - Weather	Unknown	30 minutes
[66]	Wu et al.	2018	Point	Caltrans PEMS	Apr 2014 - Jun 2015	5 minutes	None	105 minutes	45 minutes
[67]	Yao et al.	2018	Trajectory	1) New York taxi 2) New York bike	1) 1 Jan 2015 - 1 Mar 2015 2) 1 Jul 2016 - 29 Aug 2016	30 minutes	None	210 minutes	30 minutes
[68]	Zhao et al.	2018	1) Trajectory 2) Point	1) Shenzen taxi 2. LA traffic detector	1) 1 Jan 2015 - 31 Jan 2015 2) 1 March 2012 - 7 March 2012	1) 15 minutes 2) 5 minutes	None	Unknown	15, 30, 45 and 60 minutes
[69]	Pan et al.	2019	1) Trajectory 2) Point	1) Beijing taxi 2) METR-LA dataset	1) 2 Jan 2015 - 2 Jun 2015 2) 1 Mar 2012 - 30 Jun 2012	1) 1 hour 2) 5 minutes	1) Beijing POI and road network data 2) Road network data	1) 12 hours 2) 1 hour	1) 3 hours 2) 1 hour
[70]	Liang et al.	2019	Trajectory	1) Beijing taxi 2) HappyValley dataset	1) July 2013 - Oct 2013, Feb 2014 - Jun 2014, Mar 2015 - Jun 2015, Nov 2015 - Mar 2016 2) Jan 2018 - Oct 2018	1) 30 minutes 2) 1 hour	1) Weather, holidays day of week, time of day 2) Weather, holidays, ticket price, day of week time of day	Not applicable	Not applicable
[71]	He et al.	2019	Point	Hong Kong traffic	1 Jan 2017 - 30 Jun 2018	10 minutes	None	120 minutes	30, 60, 90 and 120 minutes
[72]	Do et al.	2019	Point	VicRoads Melbourne traffic	2016	5 minutes	None	180 minutes	5, 15, 30 and 60 minutes
[73]	Xie et al.	2019	Trajectory	Beilin taxi	1 Sep 2017 - 30 Nov 2017	5 minutes	Road width, road length, road category	2 hours	1 hour
[74]	Xu et al.	2019	Point	Hangzhou traffic	June 2017	15 minutes	None	45 minutes	15 minutes

data points by extending the input sequence may help in tackling the complex problem.

Some authors use data from multiple granularities and for each granularity, they may use different input sequence length. For instance, Wu and Tan [6] use the data from same day, previous day and previous week. The input sequence lengths are 75, 30 and 30 minutes respectively. Zhang *et al.* [62] use the same data selection scheme. For the same day data, they use input sequence length of 90, 120 and 150 minutes. For the day and week, they use the previous 1, 2, 3 and 4 days' and week's data.

Unfortunately, the relationship between the input sequence length and the prediction horizon is rarely explored by the literature. Most of the input sequence lengths were chosen arbitrarily without iterating through different possible values. This is because hybrid deep neural network structures take a long time to train, which makes iterating through different settings unwieldy. Despite this issue, hyperparameter search remains an important facet of deep neural network development that cannot be omitted. One possible remedy of this problem is to first use a smaller data, chosen randomly from the main dataset, to find the optimal parameter setting.

4.2 Traffic Flow Prediction – Model

The models used by the surveyed works are listed in Table 6. As observed, the two most commonly predicted values are traffic flow and traffic speed. This is because these two values are available in many popular traffic datasets. However, there are several works that deviated from these conventional data. For example, the work of Cheng *et al.* [7] predicted traffic condition, which consists of four categories: fluency, slow, congestion and extreme congestion. Zhang *et al.* [62] and Wang *et al.* [65] predicted crowd flow instead of traffic flow. Crowd flow measurements are the same as traffic flow, but they are designed for general human mobility instead of automobile mobility. In a more recent work by Liang *et al.* [70] for predicting crowd flow, a fine-grained prediction is performed using a coarser data (e.g., predicting crowd flow of different school buildings given crowd flow of the entire university area) instead of using historical data.

The column "Spatial / Temporal" in Table 6 specifies if the spatial and/or temporal factors were explicitly captured within the model. A model is said to explicitly capture spatial or temporal aspect if it satisfies at least one of the following two conditions.

TABLE 6
Traffic Flow Prediction Models

Model Category	Reference	Year	Value to Predict	Spatial / Temporal	Main Datatype	Model Subcategory
Deep neural network	[5]	2014	Traffic flow	None	Point	Deep Belief Network
	[50]	2015	Traffic flow	None	Point	Stacked Autoencoder
	[45]	2015	Traffic speed	Temporal	Point	LSTM
	[51]	2015	Traffic flow	Temporal	Point	LSTM
	[46]	2016	Traffic speed	None	Point	Deep Belief Network
	[47]	2017	Traffic speed	Temporal	Point	- Deep Belief Network - LSTM
	[61]	2017	Traffic flow	Both	Point	LSTM
	[54]	2017	Traffic speed	Both	Trajectory	CNN
	[49]	2017	Traffic flow	Both	Point	LSTM
	[65]	2018	Crowd flow	Both	Trajectory	LSTM with convolution
	[70]	2019	Crowd flow	Both	Trajectory	Residual CNN
Deep neural network, clustering and probability theory	[52]	2016	Traffic flow	None	Point	Deep Belief Network, K-means Clustering and Dempster-Shafer Theory
Deep neural network and clustering	[53]	2016	Traffic speed	Both	Trajectory	CNN and Pearson Correlation-based clustering
Deep neural network and graph theory	[48]	2017	Traffic speed	Both	Point	CNN and Graph Convolution
	[63]	2018	Traffic speed	Both	1) Point 2) Road link	LSTM and Graph Convolution
	[74]	2019	Traffic flow	Both	Point	RNN and Deepwalk
Hybrid deep neural network	[6]	2016	Traffic flow	Both	Point	LSTM, 1-D CNN and FNN
	[8]	2017	Traffic flow	Temporal	Point	LSTM and FNN
	[59]	2017	Traffic flow	Temporal	Point	LSTM and 1-D CNN
	[60]	2017	Traffic speed	Both	Point	LSTM, CNN and FNN
	[55]	2017	Traffic speed	Temporal	Point	LSTM and Stacked Autoencoder
	[4]	2017	Traffic speed	Both	Trajectory	LSTM, CNN and FNN
	[56]	2017	Traffic flow	Temporal	Point	Gated Recurrent Unit and FNN
	[62]	2017	Crowd flow	Both	Trajectory	Residual CNN and FNN
	[23]	2018	Traffic speed	Both	1) Point 2) Road link	LSTM and bidirectional LSTM
	[58]	2018	Traffic speed	Spatial	Point	Capsule network, CNN and FNN
	[64]	2018	Traffic speed	Both	Point	CNN with binary mask and FNN
	[66]	2018	Traffic flow	Both	Point	Gated Recurrent Unit, 1-D CNN and FNN
	[67]	2018	Traffic flow	Both	Trajectory	LSTM, CNN and FNN
	[71]	2019	Traffic flow	Both	Detector	Encoder-Decoder LSTM and FNN-based attention modules
	[72]	2019	Traffic flow	Both	Detector	Encoder-Decoder GRU, GRU with convolution, and attention module
Hybrid deep neural network and graph theory	[44]	2017	Traffic speed	Both	Point	Encoder-Decoder GRU and graph diffusion
	[7]	2017	Traffic condition	Both	Traffic condition	LSTM, CNN, FNN and graph based data modeling
	[57]	2018	Traffic speed	Both	Point	Encoder-Decoder LSTM, FNN and Graph CNN
	[68]	2018	Traffic speed	Both	Trajectory and point	Gated Recurrent Unit, FNN and Graph CNN
	[69]	2019	1) Traffic flow 2) Traffic speed	Both	1) Trajectory 2) Point	Encoder-Decoder GRU, FNN and Graph Attention Network
	[73]	2019	Traffic speed	Both	Trajectory	Encoder-Decoder RNN, and graph-based data modeling

- The model or at least one of its sub-components is specifically designed for capturing the spatial and/or temporal aspect.
- The data is modeled in such a way that it inherently contains the spatial and/or temporal information (e.g., using adjacency matrices to capture spatial information).

This is important as different models are proficient with capturing different aspects of the data. We will discuss the different models next.

4.2.1 RNN

Amongst the RNN-based methods, LSTM is by far the most popular one, totaling 18 out of the 37 literatures. Variants such as Gated Recurrent Unit are used in several works, but it is far less common. LSTM is the most common choice for not only capturing temporal aspect but also traffic flow prediction in general.

We speculate that this is because traffic data constitutes a temporal sequence, which fits LSTM's purpose. Additionally, most available traffic flow data is compatible with LSTM, as these traffic flow data can easily be modeled as a sequence of traffic flow readings. For instance, the traffic flow between 11:00 and 12:00 can be captured as the aggregated traffic reading for four periods, including 11:00-11:15, 11:15-11:30, 11:30-11:45, and 11:45-12:00. This data can be fed into an RNN, resulting in an RNN with four recurrences.

Basic RNN. To the best of our knowledge, the work of Ma *et al.* [45], and that of Tian and Pan [51] were the first few applications of basic LSTM. Since then, LSTM has been mostly applied in a hybrid setting, but there are still some applications of basic LSTM where the core of the model lies in how the data is modeled. For example, Fouladgar *et al.* [60] and Kang *et al.* [61] use an LSTM that takes in readings from multiple time slots as well as multiple detectors. The data is modeled in a matrix, which captures both the spatial and temporal aspects of the data.

RNN in a Hybrid Setting. As complex deep neural networks are becoming viable to train, most authors have utilized the hybrid neural network setting, which combines different neural network structures into a larger entity, to maximize the prediction performance. From Table 6, we can see that 21 out of 37 literatures utilize hybrid neural network. The popularity of the hybrid neural network structure is mainly due to its power and flexibility of utilizing the different strengths of its individual components. In a hybrid setting, RNN is used in one of the following ways:

1) Outputting features to be fed into a fusion layer.
2) Outputting features to be fed into subsequent components within the model.
3) Used as the main predictor, but with modifications to the internal structure.

The first method is the simplest because models that fall into this category usually consist of several simpler subnetworks that only interact at the final fusion layer. Wu and Tan [6] used a combination of a CNN and two LSTMs to capture spatial features, the short-term temporal feature, and the periodic temporal feature respectively. The outputs from these three networks are then fed into a FNN to fuse the features. This demonstrates one of the common usages of FNN we have discussed in Section 3.3 previously. Du *et al.* [59] used a combination of a CNN component and an LSTM component to capture spatial features and temporal features respectively. The outputs from these networks are combined to form the prediction. Another example is work [55] which used a combination of a Stacked Autoencoder to encode traffic accidents data and an LSTM to capture the temporal aspect of the data.

The second method treats LSTM as a pipeline that transforms one feature representation to another. Cheng *et al.* [7] used an LSTM to process the outputs from a CNN before passing them to a max-pooling layer. Dai *et al.* [8] performed a detrending process on the input before passing it to the LSTM layer. Yu *et al.* [4] first used a CNN to encode the spatial aspect of the data and then fed this processed information to an LSTM to learn the temporal aspect. Cui *et al.* [23] performed masking to fill in missing values in the data before passing it to a bidirectional LSTM for feature transformation and then a regular LSTM for the prediction. Zhao *et al.* [68] used a Gated Recurrent Unit which takes input from a Graph Convolution Network and outputs the predicted traffic. Yao *et al.* [67] used multiple LSTMs that represent the daily traffic features. Finally, Wu *et al.* [66] used a Gated Recurrent Unit to learn feature representation from an attention model which are then fused with the CNN spatial component. As observed, in this category of method, some preprocessing steps such as the masking of missing values can be a part of the architecture.

The third method is the most complex one, as it requires modifying the internal LSTM structure. Cui *et al.* [63] modified the LSTM calculation to include a graph convolution process as well as using a novel Real-Time Branching Learning (RTBL) which modifies the backpropagation process. Li *et al.* [44] replaced the matrix multiplication inside Gated Recurrent Unit with the diffusion convolution operation.

In addition to these methods, the encoder-decoder RNNs are also used in many recent studies. Encoder-decoder RNNs are partly inspired by autoencoders. Autoencoders are deep neural network structures that consist of two parts: the encoder that takes an input and produces a vector representation of it (usually with a smaller dimension), and the decoder that takes the vector representation and produces an approximation of the original input. In encoder-decoder RNNs, both input and output are sequences, and instead of approximating the original input, the target output is a ground-truth sequence (e.g., prediction for 5, 10, 15, 20, 25, and 30 minutes into the future). This model is used in [44], [57], [71], [69], [72] and [73], and has demonstrated state-of-the-art performance.

Other RNN Uses. Some authors have used RNNs to capture both the temporal and the spatial aspects of the data. Kang *et al.* [61] captured the temporal aspect by feeding data from multiple traffic loop detectors at once into an LSTM. Zhao *et al.* [49] used one LSTM for each traffic loop detector and incorporates an Origin Destination Correlation (ODC) matrix, which weighs how much the traffic of one loop detector's location affects another. Finally, Wang *et al.* [65] replaced the dense kernels in LSTM with convolutional ones to successfully use an LSTM to capture both the spatial and the temporal aspects of traffic data.

In addition, RNN has been used to capture the temporal aspect of the data using different granularities. As discussed in Section 3.2, RNN-based methods are commonly used to learn the temporal patterns of traffic data. However, we also mentioned that RNN-based methods are time-consuming. Consequently, RNN-based methods are not usually fed very long input sequences. Several data modeling-based approaches have been explored to mitigate this problem. The most common method is to use multiple LSTMs with each taking shorter sequences from a specific granularity. As an example, if we want to predict the traffic at 09:00 AM at December 25, one RNN can be used to capture the data from 06:00, 07:00, and 08:00 AM at December 25 (hourly granularity), one RNN can be used to capture the data from 09:00 AM at 22, 23 and 24 December (daily granularity) and one RNN can be used to capture the data from 09:00 AM at 4, 11 and 18 December (weekly granularity). This method is used by Wu and Tan [6], Yao *et al.* [67] and Wu *et al.* [66].

4.2.2 CNN

A CNN is the optimal choice for capturing the spatial aspect of the data. As mentioned in Section 3.1, CNN is able to capture the correlation between different regions in the road network. By utilizing this strength, a CNN can learn the spatial dynamics of traffic in order to improve the prediction accuracy. However, the way CNN captures this aspect strongly depends on the type of the data.

In traffic flow prediction, there are two main data types, point and trajectory, which cover the majority of the works we surveyed. The only exceptions are [63] and [23], which use road link data with point data as the main datasets. As we can treat road links as detection sites, we can regard road link data as a special type of point data. Deep neural network models, hybrid or otherwise, that are applicable for one data type are incompatible for the other without major modifications. Consequently, we categorize works related to CNN based on the type of the main datasets.

CNN With Point Data. From Table 6, we can see that most authors use point data in their work. Point data consists of traffic readings from road-installed sensors. This data is popular due to its availability and compatibility with deep neural network models; usually, point data does not require major data transformation step and can be used as it is.

For point data, spatial aspect is typically captured by collating data from multiple detection points into vectors. Sometimes, matrices can be used when capturing both the spatial and the temporal aspects. In addition, tensors can also be used when there are multiple matrices to be used all at once, such as when we are inputting the spatio-temporal traffic data from multiple days at once. These vectors/matrices/tensors are then fed as input into the network where a CNN resides.

The advantages of using point data are:

- *Common Public Data Are Available.* For instance, the Caltrans data is very commonly used in the literature. Although each work uses different subsets, the availability of one unified data source makes it easier to establish a benchmark data.
- *Data Transformation is Simpler.* To obtain an input data that contains both the temporal and the spatial trends, the common procedure is to simply collate the data into vectors/matrices/tensors.
- *Works Better for Methods That are Based on the Graph Space.* Point data often constitutes traffic detectors installed on roads, which can be easily converted to graphs; each detector site can be treated as a vertex and every two adjacent detectors define an edge. We will discuss graph-based methods in Section 4.2.5.

Although point data has multiple advantages as detailed above, it also has some limitations as listed below:

- *Almost Exclusive Highways Data.* Since traffic loop detectors are difficult and expensive to install, they are not commonly available for arterial roads.
- *Not Compatible With Methods That Conform to the Euclidean Space (e.g., 2D CNN).* This is because most point-based data are highways data where the traffic detectors are spatially organized in a line.

There are two main methods to utilize point data in a CNN. The first method is to use a 1D CNN as it is compatible with point data which are commonly organized in a line. This method is used by [6] and [59]. The second method is to capture both the spatial and the temporal aspects of the data in a 2D matrix to be fed into a CNN. That is, one axis of the matrix captures the different traffic detection sites and the other is used to capture the different time steps. This method is used by [60], [58], [53] and [64]. The work of Wu *et al.* [66] used both of these methods for different purposes.

CNN With Trajectory Data. For trajectory data, utilizing the euclidean space is common. Each trajectory needs to be mapped onto a 2D plane which represents the region (e.g., city, country) where the data resides. This region is divided into grids where each grid represents a subregion. Processing the data this way yields a matrix that represents the traffic state of a region, which can be fed into a CNN to capture the spatial aspect.

The advantages of trajectory data are:

- *Not Exclusive to Highways Data.* Trajectory data are usually GPS data, which cover both arterial roads and highways.
- *Works Better for Methods That are Based on the Euclidean Space.* After the data processing, the spatial correlation is inherently captured within the resulting 2D plane. Additionally, the resulting data transformation output is a matrix, which naturally fits 2D CNN. Finally, trajectory data usually cover city regions, which usually conform to the 2D shape.
- *Results are Easily Interpretable.* By visualizing the values assigned to each grid in the 2D map, the region's traffic flow prediction can be observed directly.

The disadvantages of trajectory data are:

- *Complex Data Transformation.* The process of mapping each trajectory point to the 2D plane is complex and time consuming.
- *Not Compatible With Methods That Model Their Data Using Graph-Based Methods.* Points in the road network can be transformed into vertices and the connections between them can be mapped to edges. This is not possible for trajectory data.

Yu *et al.* [4] mapped a road link to a 2D grid and assigned to each grid the average traffic speed of the associated road link. Zhang *et al.* [62] and Liang *et al.* [70] defined a 2D rectangular space that encompasses all the trajectory points. This space is then divided into grids. Finally, for each grid, the traffic flow for a certain period of time is calculated as the number of trajectory points that are recorded within the grid during that period. Using this modeling, the entire space can be seen as a city and the grids represent small regions within the city. Yao *et al.* [67] used a similar method as the previous, but they also modeled the traffic volume using CNN by using data of a trajectory's start and end. These four literatures represent one of the major advances of the traffic flow prediction field from the earlier classical statistical and machine learning days; an easily visualizable traffic flow prediction that utilizes trajectory data is now available due to the introduction of CNN. Although, as can be seen from Table 6, there are only a few works that utilize trajectory data. This is mainly due to the difficulty of the mapping.

Other CNN Uses. Some authors have also attempted to use CNNs to capture the temporal aspect of the data, a task usually reserved to the RNN class of methods. Ma *et al.* [54] included both the spatial and the temporal dimensions by modeling the traffic data as a tensor, where the rows represent the spatial aspect, the columns represent the temporal aspect and the depth represents the different days. They argued that using RNNs requires long input sequences which can impact training time greatly and instead applied CNN to capture both the spatial and the temporal aspects of the data. Zhang *et al.* [62] captured the temporal aspect using CNNs which are fed data from different time granularities (e.g., weekly, daily, hourly) and Yu *et al.* [48] used a one dimensional convolution on the time axis in order to capture the temporal aspect.

4.2.3 Feedforward Neural Networks

FNNs perform three main utility roles in hybrid neural networks for traffic prediction: aggregating the output of one or more subnetworks, incorporating external data (such as weather and holidays data) to the network, and as a component in the model's submodule.

FNNs as Output Aggregator. FNNs are commonly used to aggregate the output of one or more subnetwork components in a deep neural network. For instance, Wu and Tan [6] used an FNN to combine the outputs from one CNN component and two LSTM components. FNNs are also a natural component for CNNs and RNNs, since FNNs can take the output from these networks and output a smaller representation. FNNs' usage to aggregate the output of a CNN is displayed in [60], [58], [64], [67] and [4]. On the other hand, FNNs' usage to aggregate the output of an RNN is diplayed in [56], [57], and [68].

FNNs for Incorporating External Data. FNNs are also commonly used to incorporate external data to the network, because it can take inputs of an arbitrary dimensionality and perform a transformation to ensure that the dimensionality of the external data and that of the other components within the network match. Wu and Tan [6], Zhang *et al.* [62] and Yao *et al.* [67] used an FNN to perform this task.

FNNs as a Submodule Component. FNNs are often used as a component in a model's submodule, such as attention network modules. For instance, Pan *et al.* [69] used an FNN to learn features from a road network, which enables the network to learn which nodes in a road network are important. He *et al.* [71] used an FNN for the same purpose, although they do not use the graph structure.

4.2.4 Other Deep Neural Networks

Two other types of deep neural networks, Stacked Autoencoder (SAE) and Deep Belief Network (DBN), are also used in traffic flow prediction [5], [46], [47], [50], [52]. However, these models are rarely used; out of the 37 covered literatures, only 6 of them use these methods. The main contributing factor of this rarity is that SAEs and DBNs do not explicitly capture the spatial or the temporal aspect of the data and thus tend to perform worse than the neural networks that capture such aspects. This has been demonstrated through several experiments, such as in [4], [7], and [76].

In fact, SAEs and DBNs receive attention mostly at the earlier years of deep neural network for traffic flow prediction. We speculate that this is because early researchers are concerned with the computation time optimization of the training methodology. SAEs and DBNs use the greedy layerwise training method [77] to pre-train their network weights, which accelerates the training in the long run. However, as more and more complex techniques were introduced and as hardware and software optimization reduce the computational time of these complex techniques, the middling performance of SAEs and DBNs resulted in the two being phased out.

4.2.5 Other Techniques

As this paper focuses on deep neural networks, we will not discuss cases where other, non-deep-learning based methods are used as the main predictor. Rather, we discuss what other techniques have been used to assist in the prediction task alongside deep neural networks.

One of the most significant breakthroughs of recent work in deep neural network for traffic flow prediction is the graph-based methods; in particular, the graph convolution operation. When applied to road networks, graph convolution works on the graph domain while regular convolution works on the euclidean domain. However, road networks do not conform to the euclidean space as roads and highways that are close to each other may connect different parts of the city and thus have very different traffic characteristics.

Li *et al.* [44] performed a graph diffusion process based on a bidirectional graph random walk. Then, the resulting graph diffusion was used in a convolution process which is then incorporated into a Gated Recurrent Unit RNN. Cui *et al.* [63] used a similar idea of graph convolution, but instead of using the diffusion process, they proposed a method which involves calculating whether or not it is possible to reach one node from another under a certain number of time-step when the traffic is on free-flow condition. Cheng *et al.* [7] used a directed graph which represents how traffic flows between locations. Through this directed graph, it is possible to find the upstream and the downstream locations. This information is incorporated in a convolution layer. Yu *et al.* [48] modeled the traffic network as a graph and proposed a spatial graph convolutional layer. Pan *et al.* [69] modeled road network as a graph and used a graph attention network to model spatial correlations in the network. Xie *et al.* [73] used a novel component called GN block that takes a road network graph as input and outputs another graph with the same topology but different graph features. Finally, Xu *et al.* [74] used Deepwalk to transform a graph into a vector representation, which makes it easier to be incorporated into the deep neural network model.

4.3 Discussion

In this subsection, we discuss the overall trend of traffic prediction research from several different perspectives.

4.3.1 Complex Versus Simple Models

As technology advanced on both the hardware and the software front, complex deep neural network models are becoming easier to train. This has prompted researchers to combine the capabilities of multiple deep neural networks, and even add some novel components of their own creation. In Table 6, we can see that the simpler "deep neural network" category consists of papers from the earlier years of traffic prediction research while "hybrid deep neural network" and "hybrid deep neural network and graph theory" mostly contain more recent papers. Hybrid deep neural networks combine different types of simple deep neural network structures in order to combine the strengths of each. In recent works, graph theory is often applied as graphs can conform to the road structure better.

While complex models are expensive to train, their performance improvements have proven that the investment is worthwhile. For instance, Li *et al.* [44] have demonstrated that their encoder-decoder model with graph diffusion managed to outperform simple FNN and LSTM. Not only that, they also performed an ablation test to demonstrate that their novel diffusion convolution module manages to outperform simpler variations. Similarly, Do *et al.* [72] have compared their method against simple FNN, LSTM and GRU, showing similar trends. While we provide only two examples due to space constraints, we can attest that many complex hybrid deep neural network models have managed to outperform simpler deep neural network models and that many novel modules designed to capture spatial and temporal correlations (e.g., spatial and temporal attention) have resulted in further performance improvement.

4.3.2 Benchmark Model Structures

We have observed that several of the most recent and best performing models use the Encoder-Decoder RNN. In addition to the capability of processing sequential input data as regular RNN, Encoder-Decoder RNN can output sequences instead of a single result. This means that Encoder-Decoder RNNs can take input data from multiple steps and also output predictions multiple steps ahead.

To imbue Encoder-Decoder RNN with the capability to capture spatial data, most of these works also utilize graph-based methods. Graph-based methods are more appropriate for traffic data compared to the more conventional methods of dividing an area into spatial grids. The reason is that roads close to each other may connect entirely different parts of a city. It is more accurate to capture spatial correlations in terms of the connectivity, which graph-based methods provide. Encoder-Decoder RNNs and graph-based methods have been used by [44], [57], [69], and [73].

Graph-based models can be complex to implement as it requires additional data as well as data preprocessing. An alternative to this method is some sort of an attention module that can model the spatial and temporal correlations in the data. Encoder-Decoder RNNs and attention modules have been used by He *et al.* [71], and Do *et al.* [72]. Despite the complexity of Encoder-Decoder RNNs and graph-based methods, we find that this combination has shown to be very proficient at predicting future traffic and is one of the more important recent developments of traffic prediction.

Summary. In this section, we list out and categorize 37 recent literature works on deep neural network for traffic prediction. In the first subsection, we discuss the datasets, the related hyperparameters and how they affect the prediction task. In the second subsection, we discuss the models, focusing on the three main deep neural network models. Then, we described the less commonly used deep neural networks as well as other accompanying techniques. Finally, we provide a discussion section, in which we state that graph based models are one of the most important recent contributions to the traffic prediction field.

5 Challenges and Future Directions

In this section, we will first state several of the challenges outlined by Vlahogianni *et al.* in their 2014 survey paper [9] and discuss only the challenges that have been solved or partially solved using deep neural network. Afterwards, we will list several new challenges that the field of deep neural network for traffic prediction faces. Please refer to the original paper for the complete list of the ten challenges.

5.1 Existing Challenges

Developing Responsive Algorithms and Prediction Schemes. Several of the recent works have attempted to address the problem of algorithm responsiveness in the face of unexpected traffic incidents such as accidents and weather changes. This is mainly done by using weather and accidents data as additional inputs to the traffic flow prediction models.

Soua *et al.* [52] combined weather and traffic flow data using the Dempster-Shafer theory. On the other hand, Wang *et al.* [65] simply concatenated weather and traffic flow data while Zhang *et al.* [62] performed simple addition. However, these works lack ablation tests which can reveal the effectiveness of utilizing weather data.

Conversely, the work of Zhang and Kabuka [56] incorporated weather data by embedding them into the traffic flow data in their test and performed a simple ablation test, which proved that the inclusion of weather data does improve prediction performance. Additionally, Yu *et al.* [55] performed a network stimulation test to understand the effect of sudden traffic accidents.

As we can see, several authors have tested the impacts of weather and accidents in traffic flow prediction. Although several experiments have proven that the addition of these data can increase the prediction power of the models and increase their responsiveness to unexpected changes in traffic, this facet of traffic prediction has not been explored in great depth. This is due to the difficulty of incorporating these external data. Overcoming the challenge of data incorporation is the first step in utilizing weather and non-recurring incidents data in general to improve model responsiveness.

Freeway, Arterial and Network Traffic Predictions. The authors mentioned several related sub-challenges: the complexity of urban arterial traffic prediction, network-level traffic prediction and the incorporation of network dynamics on traffic prediction. We will discuss them below.

While the prediction of traffic in urban arterial roads and network-level traffic prediction are dissimilar challenges, the cause is the same: the lack of traffic detectors on urban arterial roads. This is because installing traffic detectors is costly and thus, is often done only on highways. However, the increasing amount of trajectory data has resulted in an alternative solution for network-wide prediction, as car trajectories cover both arterial and highways alike. Trajectory data is used in:, [4], [54], [62], [65] and [67].

The third challenge, incorporation of network dynamics on traffic prediction, is caused by traffic flow readings not inherently containing road network data. Therefore, this operation has to be done manually through data modeling. The most popular method to capture network data is to use graph-based methods. The literatures that cover this method are: [7], [48], [44], and [63]. Due to this ability of capturing the dynamics of road network, graph-based method is a promising future research direction.

Temporal Characteristics and Spatial Dependencies. The advances of deep neural network have brought forward two crucial network structures: RNN and CNN. These two networks can model the temporal and spatial patterns of the data, respectively. Please refer to Sections 4.2.1 and 4.2.2 for a more detailed description of how these models are used to capture temporal characteristics and spatial dependencies.

Explanatory Power, Associations and Causality. Neural network's prominence in traffic flow prediction can be attributed to the model's flexibility. This is because the functional form of neural network models is approximated via learning, as opposed to classical statistical models which assume the functional form a priori [24]. Consequently, neural network models' internal parameters are rarely explored because they are hard to interpret as their focus is mostly on raw prediction performance rather than interpretability.

Performing explanatory analyses on neural networks may uncover useful traffic patterns. Li *et al.* [44] observed the traffic diffusion along the road network and the correlation between several adjacent traffic sensors. Cui *et al.* [63] visualized the network weights pertaining to different detector sites to find key road segments in the traffic network. Cheng *et al.* [7] visualized the attention weights of upstream and downstream stations to observe how traffic flow moves across several traffic stations.

While neural networks have proven to be a very effective prediction model, they are infamously known as black-box models; models that are difficult to dissect and explain. Although the aforementioned authors managed to explain the traffic phenomena to some degree, their observations are mostly limited to the spatial aspect; observing how the traffic at one site affects another and how traffic propagates across the road network. To the best of our knowledge, there is no work that observes other aspects of the prediction, such as the dynamics of abrupt weather changes and accidents.

5.2 Future Challenges

The power of deep neural network as prediction models has brought forward new challenges, both for the models and for the field as a whole. We will now discuss these challenges.

A). Lack of a Benchmark Dataset. The availability of a wide range of traffic data supports traffic prediction. However, this availability also poses a challenge to comparative work. Due to the fact that different works use different datasets, it is very hard to assess the relative performance of different state-of-the-art models. The Caltrans data is the closest to a benchmark dataset, as it is used by 14 out of 37 literatures we have covered. However, different works use different subsets of the Caltrans from different periods of time and from different traffic detector sites.

Choosing a subset of data within a larger dataset also poses a challenge. As temporal and spatial correlation affects traffic greatly, the period of the data and the traffic detector locations become important considerations. For instance, when using data that covers a period of less than a year, there is a risk of not capturing the seasonal effects on traffic, and when using only weekdays data, the models cannot learn weekend traffic well. For the spatial aspect, the choice of roads or highways can greatly affect the traffic flow as metropolitan roads have significantly busier traffic compared to rural areas, and long interstate highways tend to cover both rural and metropolitan areas. Models that are trained on a certain traffic condition may not perform well when used to predict traffic on significantly different traffic.

Both point and trajectory data have their advantages and disadvantages. Point data generally comes from traffic detectors installed by the transportation bureau. Consequently, the system is well-established, resulting in better temporal coverage. However, as traffic detectors are costly to install, they are mostly limited to highways. Conversely, trajectory data has a more general spatial coverage as drivers pass through arterial, urban and highway roads alike. However, the temporal coverage is limited, ranging from a month [67], [68] to several months [4], [62], [67] and up to one year [53], [65], compared to the Caltrans data, for instance, which contains more than five years' worth of data for its detectors.

For deep neural network models to perform well on real applications, the dataset needs to mimic real data. Therefore, it is important for benchmark datasets to cover enough time frame and locations so that the models can generalize well to any traffic situations. To overcome this challenge, the following criteria are important:

- The data covers both urban and rural areas.
- The data covers both weekdays and weekends.
- The data covers all hours of the day.
- The temporal range is at least one year.

The installation of traffic detectors is expensive, and point data's spatial limitation is difficult to address. Therefore, we recommend focusing on trajectory data. Floating car data collected from GPS is the most widespread and efficient source of trajectory data. However, researchers must take into account the required preprocessing to use trajectory data for traffic prediction.

B). Difficulty of Incorporating External Information With Traffic Data. In traffic prediction, commonly used external information include weather, accidents, events, day of the week, time of the day and social media data. While the inclusion of day of the week and time of the day is relatively simple, data that are bound to a specific geographical coordinate or a geographical area is difficult to incorporate with traffic data. This is because the process requires the coordinates of detection points (in the case of point data) or trajectory points (in the case of trajectory data) to be mapped to the secondary data. A benchmark data that covers a specific area within a specific period, complete with relevant secondary data will greatly benefit the traffic flow field. We recommend the following sequence of actions:

1) Establish a benchmark dataset that has sufficient spatial and temporal coverage based on the requirements mentioned in the previous challenge.
2) Add day of the week and time of day data by concatenating them with the traffic reading data.
3) Add geographical-related data, such as weather and accidents data to every traffic data reading. For instance, one reading at a particular time stamp and location will have both the traffic flow, current weather and accident type, if any accident occurs at the location.

C). Online Learning. With the widespread installation of traffic loop detectors, traffic data will continuously grow. In this setting where new data is incrementally added, traffic trends will shift over time. This is applicable even for the same traffic detector site. This idea is called concept drift and it causes the relationship between the input and output data to change over time, rendering models that are trained on past data to degrade in performance on present and future data.

One way to mitigate this problem is to incrementally update the prediction model with new data in real-time, in a process often called online learning. However, to the best of our knowledge, there is no work that explores online learning in the traffic prediction domain. This can be attributed to the time complexity of training hybrid deep neural network model and the lack of attention to the concept drift problem. Online learning is a promising subtopic to explore in the field of traffic prediction as this will ensure that complex deep neural network models are always up-to-date. Experiments that seek to identify the viability of online training will need to take into account the following factors:

- The frequency of which the deep neural network models need to be retrained. Practitioners need to ask the question "How often do we need to update our prediction model to ensure that it is always up-to-date?"
- The number of data points required for the update. This is affected by the frequency, and has to reflect real life scenario. Practitioners need to ask the questions "How much data can we acquire during a certain period?" and "How long will it take to collect and preprocess the data to fit it into the prediction models?"
- The time required for the model to be re-trained using the specified number of data points and whether or not it is suitable for real life scenario. Practitioners need to ask the question "With the available amount of data, will the training of the model be fast enough such that daily operations are not hindered?"

D). Using Graph-Based Methods to Capture Spatial Aspect of the Data. As we have discussed in Section 4.2.5, graph based methods are a promising development of traffic flow prediction, because they naturally conform to traffic dynamics. However, the difficulty lies in the data requirement and the additional preprocessing step. The road topology data, which captures how different traffic detection sites are connected by roads, is usually unavailable and has to be manually curated. While this challenge is significant, it is important to measure and understand how well graph-based methods improve the traffic prediction performance when combined with deep neural networks.

E). Exploring Other Traffic Prediction Tasks. Currently, the Intelligent Transportation Systems (ITS) field greatly focuses on traffic flow prediction, neglecting the other traffic prediction tasks. Exploring these subproblems may bring new insights that are able to help the main traffic prediction task. As we mentioned before, deep neural network models are black-box models. Models that are trained on the traffic flow prediction may not be able to explain the intricacies of traffic patterns. Additionally, each of the subproblems is interesting by itself as its results can be directly used by drivers and traffic management bureau alike to make educated decisions. One example of these prediction tasks is traffic congestion analysis. Knowing how traffic congestion moves throughout the network can assist in the traffic prediction task.

F). Lack of Up-to-Date Experimental Evaluation. The introduction of deep neural network libraries such as Keras [12], PyTorch [13] and TensorFlow [14] has simplified the implementation of complex hybrid deep neural network models. As we have observed, this has resulted in numerous unique hybrid structures, each focusing on specific ideas to improve prediction performance. However, there is a lack of up-to-date and comprehensive experimental evaluation, making it difficult to assess how promising these specific ideas are.

Experimental evaluation in traffic flow prediction is complex due to two factors. The first is the lack of benchmark dataset, a problem that we have discussed previously. The second is the lack of code availability. One might attempt to recreate the model from the author's description. However, while the deep neural network aspect can be recreated relatively easily, novel components such as graph diffusion are difficult to build in a way that is faithful to the source material.

This lack of experimental evaluation is perhaps the biggest challenge that the traffic flow prediction community faces. Addressing this problem will enable practitioners to easily identify the effectiveness of new ideas in improving prediction performance, model efficiency, and the overall applicability of deep neural network models in real-time traffic prediction applications. A benchmark experimental evaluation needs to take into account the following insights:

- The impact of each model's novel ideas to the prediction power, particularly for models that use a similar network structure.

- The impact of using a certain neural network type such as CNN and RNN.
- The viability in real life applications with respect to the retraining time. That is, online learning using a realistically sized batch of data, e.g., data from one week.
- The impact of using external information such as weather and accidents data. This can be observed by performing an ablation test on models that utilize these external information.

G). Applying Emerging Techniques There are several emerging techniques that have been applied to the problem of traffic flow prediction. However, as these technologies are still in their infancy, they are much rarer compared to the more conventional deep neural network structures discussed in the previous sections. Two promising new techniques are Transformers [78] and Generative Adversarial Networks (GAN) [79].

Transformers are similar to encoder-decoder RNNs in that they take sequences as inputs and outputs sequences. The difference is that Transformers are designed with attention mechanisms in mind and can be parallelized. The original paper by Vaswani *et al.* [78] applied Transformers to machine translation, but it has since been applied to traffic flow prediction by Xu *et al.* [80].

Generative Adversarial Networks consist of two neural networks that are trained to compete with each other. The two networks are generative networks, designed to capture the data distribution, and discriminative network, which judges whether a given sample came from the true data or from the distribution generated by the generative network [79]. This method has been used by Liang *et al.* [81], which use LSTMs for both the generative and discriminative network, and by Lin *et al.* [82], where a GAN is used to enable traffic flow prediction that is more robust to outliers. Zhang *et al.* [83] combine GAN with graph CNN, and use sequence-to-sequence autoencoder for the generative network.

While state-of-the-art models that commonly use encoder-decoder LSTM combined with graph-based methods, have achieved excellent performance, these promising new techniques may be able to further improve the performance of traffic flow prediction.

6 Conclusion

Traffic flow prediction is one of the easiest and cheapest measures to address traffic congestion. In this work, we have explored how the field of traffic flow prediction had evolved over the time from classical statistical models, to machine learning models and finally to deep neural network models; described the common deep neural network structures, how they work and how they are able to learn specific features from traffic data; listed out and compared the numerous deep neural network for traffic flow prediction literatures; and identified the existing and future challenges faced by the traffic flow prediction field.

We believe that the future of the traffic flow prediction field lies on determining a more standardized approach that ensures that the significance of every novel idea can be identified. The first step is to establish a comprehensive benchmark dataset that enables the multitude of traffic factors to be explored; not only from the spatiotemporal side, but incorporating social media data, weather data, accidents data and many other external data that might affect traffic prediction. Then, the next step is to provide more transparency in this research field. Implementation details and publicly accessible codes will be necessary. The final step is then to provide readers and practitioners alike with an up-to-date, thorough snapshot of the current advances in the field. Continuous survey and especially experimental evaluation work contribute to this goal.

We hope that the advances of the traffic flow prediction field will inspire confidence and eventual widespread implementation of real-time prediction systems that can directly contribute to the improvements of traffic condition worldwide.

Acknowledgments

This work was partially supported by ARC under Grants DP200102611, DP180102050, and LP180100114, a Google Faculty Research Award, and the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore.

References

[1] Australian Government Bureau of Infrastructure, Transport and Regional Economics, (2015), *Information Sheet 74 - Traffic and congestion cost trends for Australian capital cities*, viewed Mar. 18, 2020.

[2] BBC News, "Singapore to freeze car numbers," 2017, Accessed: Nov. 20, 2018. [Online]. Available: https://www.bbc.com/news/business-41730778

[3] J. Carnegie and A. M. Voorhees, *The Cost of Roadway Construction, Operations and Maintenance in New Jersey: Phase 1 Final Report*, pp. 557–566, 2016.

[4] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, pp. 1–16, 2017.

[5] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[6] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," 2016, *arXiv:1612.01022*.

[7] X. Cheng, R. Zhang, J. Zhou, and W. Xu, "Deeptransport: Learning spatial-temporal dependency for traffic condition forecasting," in *Proc. Int. Joint Conf. Neural Netw.*, 2017, vol. 1709.09585, pp. 1–8.

[8] X. Dai, R. Fu, Y. Lin, L. Li, and F. Wang, "Deeptrend: A deep hierarchical neural network for traffic flow prediction," 2017, *arXiv:1707.03213*.

[9] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. Part C: Emerg. Technologies*, vol. 43, pp. 3–19, 2014.

[10] A. M. Nagy and V. Simon, "Survey on traffic prediction in smart cities," *Pervasive Mobile Comput.*, vol. 50, pp. 148–163, 2018.

[11] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.

[12] F. Chollet *et al.*, "Keras," 2015. [Online]. Available: https://keras.io

[13] A. Paszke *et al.*, "Automatic differentiation in pytorch," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017, pp. 1–4.

[14] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[15] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time series data by using box-jenkins techniques," *Transp. Res. Rec.*, vol. 773, pp. 1–9, 01 1979.

[16] M. Levin and Y.-D. Tsao, "On forecasting freeway occupancies and volumes (abridgment)," *Transp. Res. Rec.*, no. 773, pp. 47–49, 1980.

[17] S. Lee and D. Fambro, "Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting," *Transp. Res. Rec.*, vol. 1678, pp. 179–188, Nov. 1999.

[18] B. Williams, "Multivariate vehicular traffic flow prediction: Evaluation of arimax modeling," *J. Transp. Res. Board*, no. 1776, pp. 194–200, 2001.

[19] B. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, pp. 664–672, Nov. 2003.

[20] Y. Kamarianakis and P. Prastacos, "Forecasting traffic flow conditions in an urban network : Comparison of multivariate and univariate approaches," *J. Transp. Res. Board*, vol. 1857, pp. 74–84, 2003.

[21] M. Karlaftis and E. Vlahogianni, "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights," *Transp. Res. Part C: Emerg. Technologies*, vol. 19, no. 3, pp. 387–399, 2011.

[22] Y. Li and C. Shahabi, "A brief overview of machine learning methods for short-term traffic forecasting and future directions," *SIGSPATIAL Special*, vol. 10, pp. 3–9, 2018.

[23] Z. Cui, R. Ke, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," 2018, *arXiv:1801.02143*.

[24] B. Warner and M. Misra, "Understanding neural networks as statistical tools," *Amer. Statistician*, vol. 50, no. 4, pp. 284–293, 1996.

[25] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term traffic forecasting: Overview of objectives and methods," *Transp. Rev.*, vol. 24, no. 5, pp. 533–557, 2004.

[26] M. Dougherty, H. Kirby, and R. Boyle, "The use of neural networks to recognise and predict traffic congestion," *Traffic Eng. Control*, vol. 34, pp. 311–314, 1993.

[27] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach," *Transp. Res. Part C: Emerg. Technologies*, vol. 13, no. 3, pp. 211–234, 2005.

[28] W. Zheng and D.-H. Lee, "Short-term freeway traffic flow prediction: Bayesian combined neural network approach," *J. Transp. Eng.*, vol. 132, pp. 114–121, 2006.

[29] M. Zhong, S. Sharma, and P. Lingras, "Short-term traffic prediction on different types of roads with genetically designed regression and time delay neural network models," *J. Comput. Civil Eng.*, vol. 19, no. 1, pp. 94–103, 2005.

[30] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, "Neural network based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg Marquardt algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, pp. 644–654, Jun. 2012.

[31] G. A. Davis and N. L. Nihan, "Nonparametric regression and short-term freeway traffic forecasting," *J. Transp. Eng.*, vol. 117, pp. 178–188, Mar. 1991.

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

[33] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. Part C: Emerg. Technol.*, vol. 62, pp. 21–34, 2016.

[34] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Syst. With Appl.*, vol. 36, no. 3, pp. 6164–6173, Apr. 2009.

[35] H. Su, L. Zhang, and S. Yu, "Short-term traffic flow prediction based on incremental support vector regression," in *Proc. Int. Conf. Natural Comput.*, 2007, vol. 1, pp. 640–645.

[36] X. Jin, Y. Zhang, and D. Yao, "Simultaneously prediction of network traffic flow based on PCA-SVR," in *Proc. Int. Conf. Neural Netw.*, 2007, pp. 1022–1031.

[37] H. Hu, G. Li, Z. Bao, Y. Cui, and J. Feng, "Crowdsourcing-based real-time urban traffic speed estimation: From trends to speeds," in *Proc. Int. Conf. Data Eng.*, 2016, pp. 883–894.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] F. A. Gers, F. Cummins, and J. Schmidhuber, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, pp. 2451–2471, 2000.

[40] Y. Sun, L. Li, Z. Xie, Q. Xie, X. Li, and G. Xu, "Co-training an improved recurrent neural network with probability statistic models for named entity recognition," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2017, pp. 545–555.

[41] J. Rao, F. Türe, H. He, O. Jojic, and J. Lin, "Talking to your TV: context-aware voice search with hierarchical recurrent neural networks," in *Proc. Conf. Inf. Knowl. Manage.*, 2017, pp. 557–566.

[42] D. Eck and J. Schmidhuber, "A first look at music composition using LSTM recurrent neural networks," IDSIA-07-02, Instituto Dalle Molle di studi sull' intelligenza artificiale, Switzerland, 2002.

[43] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3156–3164.

[44] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *Proc. ICLR*, 2018, vol. 1707.01926.

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

[46] Y. Jia, J. Wu, and Y. Du, "Traffic speed prediction using deep learning method," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst.*, 2016, pp. 1217–1222.

[47] Y. Jia, J. Wu, M. Ben-Akiva, R. Seshadri, and Y. Du, "Rainfall-integrated traffic speed prediction using deep learning method," *IET Intell. Transp. Syst.*, vol. 11, no. 9, pp. 531–536, 2017.

[48] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional neural network: A deep learning framework for traffic forecasting," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 3634–3640.

[49] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, 2017.

[50] L. Yisheng, Y. Duan, and W. Kang, "Traffic flow prediction with big data : A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[51] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *Proc. IEEE Int. Conf. Smart City/SocialCom/SustainCom*, 2015, pp. 153–158.

[52] R. Soua, A. Koesdwiady, and F. Karray, "Big-data-generated traffic flow prediction using deep learning and dempster-shafer theory," in *Proc. Int. Joint Conf. Neural Netw.*, 2016, pp. 3195–3202.

[53] J. Wang, Q. Gu, J. Wu, G. Liu, and Z. Xiong, "Traffic speed prediction and congestion source exploration: A deep learning method," in *Proc. Int. Conf. Data Mining*, 2016, pp. 499–508.

[54] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, Apr. 2017, Art. no. 818.

[55] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: A generic approach for extreme condition traffic forecasting," in *Proc. SIAM Int. Conf. Data Mining*, 2017, pp. 777–785.

[56] D. Zhang and M. R. Kabuka, "Combining weather condition data to predict traffic flow: A GRU based deep learning approach," in *Proc. Int. Conf. Big Data Intell. Comput.*, 2017, pp. 1216–1219.

[57] B. Liao *et al.*, "Deep sequence learning with auxiliary information for traffic prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 537–546.

[58] Y. Kim, P. Wang, Y. Zhu, and L. Mihaylova, "A capsule network for traffic speed prediction in complex road networks," *Sensor Data Fusion: Trends, Solutions, and Applications*, 2018, pp. 1–7.

[59] S. Du, T. Li, X. Gong, Y. Yang, and S. J. Horng, "Traffic flow forecasting based on hybrid deep learning framework," in *Int. Conf. Intell. Syst. Knowl. Eng.*, 2017, pp. 1–6.

[60] M. Fouladgar, M. Parchami, R. Elmasri, and A. Ghaderi, "Scalable deep traffic flow neural networks for urban traffic congestion prediction," in *Proc. Int. Joint Conf. Neural Netw.*, 2017, pp. 2251–2258.

[61] D. Kang, L. Yisheng, and Y.-Y. Chen, "Short-term traffic flow prediction with LSTM recurrent neural network," in *Proc. Intell. Transp. Syst.*, 2017, pp. 1–6.

[62] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[63] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "High-order graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *Proc. Trans. Res. Board 98th Annu. Meeting*, 2018, Art. no. 6.

[64] S. Ren, B. Yang, L. Zhang, and Z. Li, "Traffic speed prediction with convolutional neural network adapted for non-linear spatio-temporal dynamics," *Proc. ACM SIGSPATIAL Int. Workshop*, 2018, pp. 32–41.

[65] L. Wang, X. Geng, X. Ma, F. Liu, and Q. Yang, "Crowd flow prediction by deep spatio-temporal transfer learning," 2018, *arXiv:1802.00386*.

[66] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transp. Res. Part C: Emerg. Technol.*, vol. 90, pp. 166–180, 2018.

[67] H. Yao, X. Tang, H. Wei, G. Zheng, Y. Yu, and Z. Li, "Modeling spatial-temporal dynamics for traffic prediction," *CoRR*, vol. 1803.01254, 2018.

[68] L. Zhao, Y. Song, M. Deng, and H. Li, "Temporal graph convolutional network for urban traffic flow prediction method," 2018, *arXiv:1811.05320*.

[69] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1720–1730.

[70] Y. Liang *et al., "Urbanfm: Inferring fine-grained urban flows," in* Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 3132–3142*.

[71] Z. He, C.-Y. Chow, and J.-D. Zhang, "STANN: A spatio-temporal attentive neural network for traffic prediction," *IEEE Access*, vol. 7, pp. 4795–4806, 2019.

[72] L. Do, H. Vu, B. Vo, Z. Liu, and D. Phung, "An effective spatial-temporal attention based neural network for traffic flow prediction," *Transp. Res. Part C: Emerg. Technol.*, vol. 108, pp. 12–28, 2019.

[73] Z. Xie, W. Lv, S. Huang, Z. Lu, B. Du, and R. Huang, "Sequential graph neural network for urban road traffic speed prediction," *IEEE Access*, vol. 8, pp. 63 349–63 358, 2020.

[74] D. Xu, H. Dai, Y. Wang, P. Peng, Q. Xuan, and H. Guo, "Road traffic state prediction based on a graph embedding recurrent neural network under the scats," *Chaos: An Interdisciplinary J. Nonlinear Sci.*, vol. 29, pp. 1–10, 2019.

[75] Transportation Research Board, *Highway Capacity Manual*, Washington DC, Transportation Research Board, 2010.

[76] H. Zhang, H. Wu, W. Sun, and B. Zheng, "Deeptravel: A neural network based travel time estimation model with auxiliary supervision," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 3655–3661.

[77] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Neural Inf. Process. Syst.*, 2006, pp. 153–160.

[78] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[79] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

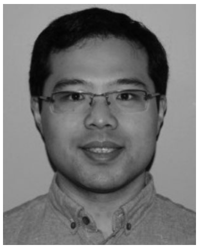
[80] M. Xu *et al.*, "Spatial-temporal transformer networks for traffic flow forecasting," 2020, *arXiv:2001.02908*.

[81] Y. Liang, Z. Cui, Y. Tian, H. Chen, and Y. Wang, "A deep generative adversarial architecture for network-wide spatial-temporal traffic state estimation," *Transp. Res. Rec.: J. Transp. Res. Board*, vol. 2672, pp. 87–105, 2018.

[82] Y. Lin, X. Dai, and L. Li, "Pattern sensitive prediction of traffic flow based on generative adversarial framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, pp. 2395–2400, Jun. 2019.

[83] Y. Zhang, S. Wang, B. Chen, and J. Cao, "GCGAN: Generative adversarial nets with graph CNN for network-scale traffic prediction," in *Proc. Int. Joint Conf. Neural Netw.*, 2019, pp. 1–8.

David Alexander Tedjopurnomo received the bachelor of information technology degree from The University of Newcastle, in 2014, and the master of information technology degree from The University of Melbourne, in 2017, he is currently working toward the PhD degree in computer science at RMIT University. His research interests include traffic prediction, trajectory queries, and deep learning for spatiotemporal data analytics.



Zhifeng Bao received the PhD degree in computer science from the National University of Singapore, in 2011, as the winner of the Best PhD Thesis in the school of computing. He is currently an associate professor at RMIT University. He is also an Honorary senior fellow with the University of Melbourne in Australia. His current research interests include data usability, spatial database, data integration, and data visualization.

Baihua Zheng received the PhD degree in computer science from the Hong Kong University of Science and Technology, China, in 2003. She is currently a professor with the School of Information Systems, Singapore Management University, Singapore. Her research interests include mobile/pervasive computing, spatial databases, and big data analytics.

Farhana Murtaza Choudhury received the PhD degree in computer science from RMIT University, in 2017. She is currently a lecturer with the University of Melbourne. Her current research interests include spatial databases, data visualization, trajectory queries, and applying machine learning techniques to solve spatial problems.

A. K. Qin (Senior Member, IEEE) received the BEng. degree FROM Southeast University, China, in 2001, and the PhD degree from Nanyang Technology University, Singapore, in 2007. From 2007 to 2012, he worked first at the University of Waterloo (Canada) and then at INRIA (France). Since 2013, he was a vice-chancellor's research fellow, lecturer and senior lecturer at RMIT University, Australia. In February 2017, he joined the Swinburne University of Technology, Australia as an associate professor. Currently, he is leading Swinburne's Intelligent Data Analytics Lab and Machine Learning and Intelligent Optimization (MLIO) Research Group. His major research interests include evolutionary computation, machine learning, computer vision, GPU computing, services computing, and mobile computing. He won the 2012 IEEE Transactions on Evolutionary Computation Outstanding Paper Award and the Overall Best Paper Award at the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES 2014). He is now co-chairing the IEEE Emergent Technologies Task Forces on "Collaborative Learning and Optimization" and "Multitask Learning and Multitask Optimization".