Short-term Traffic Prediction with Deep Neural Networks: A Survey

Kyungeun Lee[1], Moonjung Eo[1], Euna Jung[1], Yoonjin Yoon[2], and Wonjong Rhee[1,*]

[1]Department of Intelligence and Information, Seoul National University, Seoul, Korea

[2]Department of Civil and Environmental Engineering, KAIST, Daejeon, Korea

[*]Corresponding author: Wonjong Rhee, wrhee@snu.ac.kr

Abstract

In modern transportation systems, an enormous amount of traffic data is generated every day. This has led to rapid progress in short-term traffic prediction (STTP), in which deep learning methods have recently been applied. In traffic networks with complex spatiotemporal relationships, deep neural networks (DNNs) often perform well because they are capable of automatically extracting the most important features and patterns. In this study, we survey recent STTP studies applying deep networks from four perspectives. 1) We summarize input data representation methods according to the number and type of spatial and temporal dependencies involved. 2) We briefly explain a wide range of DNN techniques from the earliest networks, including Restricted Boltzmann Machines, to the most recent, including graph-based and meta-learning networks. 3) We summarize previous STTP studies in terms of the type of DNN techniques, application area, dataset and code availability, and the type of the represented spatiotemporal dependencies. 4) We compile public traffic datasets that are popular and can be used as the standard benchmarks. Finally, we suggest challenging issues and possible future research directions in STTP.

Keywords— Deep neural network (DNN), short-term traffic prediction (STTP), survey

1 Introduction

Advances in transportation systems have resulted in the generation of a large amount of traffic data from various sources [1–3]. In everyday life, GPS sensors installed in smartphones carried by millions of people can collect *crowd flow* data. Furthermore, taximeters and bus card readers can collect *crowd demand* data, and vehicle loop detectors can collect *traffic flow* or *speed* data. In the mean time, Deep Neural Networks (DNNs) have achieved promising performance improvements in various application areas. They can classify images into thousands of classes [4,5] as well as recognize human speech [6,7], with only small errors. Owing to the availability of large traffic datasets and the advances in DNN techniques, DNNs have recently been applied to Short-Term Traffic Prediction (STTP).

Compared with classic machine learning algorithms, such as linear regression and Auto-Regressive Integrated Moving Averages (ARIMA), DNNs can fit a wide variety of functions. A typical DNN model has hundreds of thousands or even tens of millions of parameters that are learnt during training. Such a large number of parameters would result in overfitting in classical machine learning algorithms. By contrast, DNNs are known to avoid overfitting with a reasonable amount of data [8]. Thus, based on large datasets, DNNs could learn any highly complex non-linear functions without serious overfitting issues.

Although DNNs can automatically extract the underlying features, in practice, they are fed with appropriate inputs, the representations of which can be matched with the intended *inductive bias* and deep network architecture [9]. *Inductive bias* induces DNNs to prioritize one solution over another by imposing constraints on relationships and interactions among entities in a learning process based on prior knowledge. Determining the type of inductive bias to be induced affects the DNN architecture design, and therefore it also affects the final prediction performance. For instance, in [10], one of the most famous priors for the spatial dependency was proposed: *"Everything is related to everything else, but near things are more related than distant things"*. If this prior knowledge is to be used as DNN inductive bias, we should investigate *which input data format* could properly represent locality, and *which DNN type* could effectively capture the represented locality. Intuitively, grid-based image-like representations, such as satellite photographs, may be able to represent locality, as they convert proximate regions into adjacent pixels. In addition, convolutional layers can effectively capture image locality by training adjacent pixels using a single local filter.

In machine learning research, standard benchmark datasets are often used so that algorithms can be easily compared, and studies easily combined. For instance, CIFAR-10 [11]/ CIFAR-100 [12] and ImageNet [13] are known as the benchmark datasets for visual tasks, and they have greatly contributed to the recent progress in this area. In STTP, however, there is *no* standard benchmark dataset. If there were such a dataset, related research would be significantly accelerated because findings would be easy to share and integrate. Accordingly, we summarize public datasets that have been widely used in previous studies. In addition, we summarize papers with publicly available code.

The remainder of this paper is organized as follows. Section 2 presents input data representation methods for spatial and temporal dependencies of traffic networks. Section 3 overviews deep learning techniques in chronological order. Section 4 summarizes previous work according to the application area. Section 5 introduces open datasets important for generalizability. Section 6 discusses current challenges and possible research directions in STTP.

2 Input Data Representation Methods

Short-term traffic prediction is based on spatiotemporal information of a traffic network consisting of several spatial units (e.g., links, roads segments, and regions). Early studies assumed that it suffices to represent spatiotemporal information as independent features without considering any dependencies between spatial or temporal units. For example, to predict future crowd flow, [14] generated independent features, such as inbound or outbound, holiday or not, weekday, and hours.

However, a traffic network includes complex and non-linear relationships between spatial and/or temporal units. In theory, deep networks can learn any latent relationship between input units in the dataset. In practice, however, the actual performance after training is heavily dependent on the input data representation, which is therefore an important matter. In addition, some networks impose important relational inductive biases, which can be effectively provided through a specific type of input data representation [9]. For instance, convolutional layers provide locality and translation invariance as a relative inductive bias. Thus, input data should be arranged to make local units proximate using grid representation. Otherwise, recurrent layers carry sequentiality and temporal invariance as a relative inductive bias. Thus, input data should be aligned in a temporal order. Deep network architectures are explained in more detail in Section 3.

Herein, we summarize the representation of spatial and temporal dependencies in STTP according to the *number* and *type* of dependencies that are manipulated in a sample. In general, one dependency corresponds to one data dimension. We note that external information (e.g., weather conditions, temperature, and wind speed) is not mentioned because it is considered sufficient to represent such information as independent features only in most studies.

2.1 Representing Spatial Dependencies

2.1.1 Stacked Vector

To represent a spatial dependency, we can simply stack the data of spatial units into a single vector according to a predefined rule depending on domain knowledge or personal preference. For instance, to represent the traffic network $L = \{L_1, , L_{10}\}$ (L_i indicates a road segment), as shown in Fig. 1a, we can simply stack the road segments in clockwise connection order. Then, the final representation is $[V(L_1), V(L_2), , V(L_{10})]$, where $V(L_i)$ is the traffic value of L_i at a specific time. We found some general representation rules, such as connection order, instream/outstream(inflow/outflow) order, and random listing. This method is quite simple and useful when the network has a powerful dependency. However, it does not provide general procedures for determining the representation in general networks. As a simple example, it is not trivial to determine the dependency representation rule in the case of Fig. 1b.
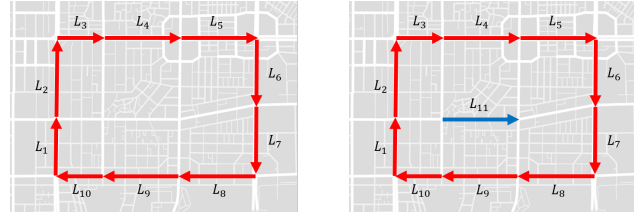
(a) Example of traffic network in which it is easy to stack the road segments in clockwise connection order.

(b) Example of traffic network in which a stacking method for input data representation is not easily applied.

Figure 1: Example of stacked-vector method for input data representation.

Table 1: Summary of representation methods. N is the number of spatial or temporal dependencies that are considered for forming the input data representation.

Spatial Dependency			Temporal Dependency		
N	Representation Methods	Abbr.	N	Representation Methods	Abbr.
1	Stacked vector (random)	1-SR	1	Sequentiality	1-S
	Stacked vector (stream flow)	1-SF			
	Graph (considering single dependency)	1-Gr			
2	Grid representation (coordinates)	2-G	2	Sequentiality and periodicity (weekly)	2-SW
				Sequentiality and periodicity (daily)	2-SD
3	Grid representation (including inflow/outflow)	3-GIO	3	Sequentiality and periodicity (weekly and daily)	3-SWD
	Graph (considering three dependencies)	3-Gr			

2.1.2 Grid representation

It is natural to regard spatial information as 2-dimensional Euclidean data consisting of latitude and longitude. Thus, without any modification, we can provide raw information to deep networks as an image-like representation. This method is also known as grid-based map segmentation or grid map. For instance, in the traffic network $L = \{L_1, , L_{17}\}$ (L_i indicates a road segment), as shown in

(a) Example of traffic network. (b) Grid representation applied to the input data of (a). (c) Example of traffic network not suitable for grid representation. (d) Example of traffic network, including surrounding area information, suitable for the graph representation method.
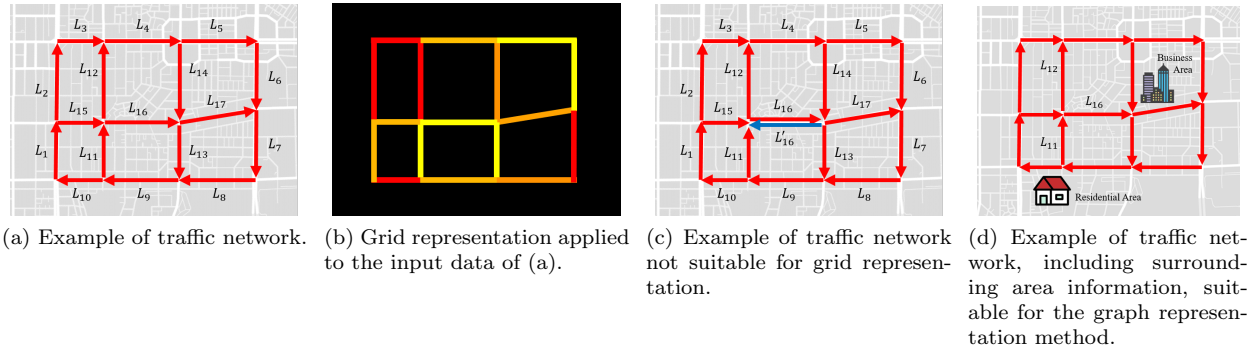
Figure 2: (a)–(c) Examples of the grid representation method, and (d) is an example of needing to apply the graph representation method. Best viewed in color.

Fig. 2a, the final representation is the 2-d image-like representation shown in Fig. 2b. Each pixel has a value (represented as a color in Fig. 2b) depending on the traffic value at a specific time. In addition, recent studies have extended this method so that there are two individual channels by the flow direction (inflow/outflow). In this case, three spatial dependencies can be included in one sample.

This method is quite intuitive, but its disadvantage is the inefficiency of the representation. As shown in Fig.2b, most of the grid points do not match any road links, and zeros should be inserted to indicate that there are no road links. This inefficiency can worsen if higher resolution is required. The low resolution can also reduce the representation efficiency because several spatial units might correspond to the same grid point. In this case, we only can represent one representative value like the average value of the units. It makes the represented input data lose part of the original and essential information. This low resolution problem can be a serious issue, particularly when a pair of road links in opposite directions should be represented, as shown in Fig. 2c.

2.1.3 Graphs

Unlike image data, traffic network data have more complicated spatial dependencies (which cannot be explained only by Euclidean information) resulting from the connection order, non-regular traffic signals, and abrupt events or accidents. To consider this type of non-Euclidean dependencies, we can use graph representations. A simple example of a useful non-Euclidean spatial dependency is shown in Fig. 2d, where some spatial unit labels are omitted for clarity. The labels are the same as in Figs. 1 and 2a. Here, we intend to model the relation of the traffic value of L_{11} to the other directly connected segments, that is, L_{12} and L_{16}. Regarding commute hours, even though L_{11} and L_{16} share the driving route from the residential to the business area, L_{12} does not. Thus, it would be natural to speculate that L_{11} is more related to L_{16} than L_{12}, even though both are equally proximate to L_{11}.

A graph consists of nodes (vertices) and edges, and each node corresponds to a spatial unit. In a graph representation of complex pairwise information, an $N \times N$ matrix can be used, where N is the number of nodes. This matrix is typically called an *adjacency matrix*, and its (i, j) element represents the pairwise relationship between spatial units i and j. Examples of complex (non-Euclidean) pairwise information include distance, connectivity, and other spatial relationships. Graph representation has recently gained increasing popularity [15–28] because the $N \times N$ matrix is an efficient means of representing spatial relationships, and the resulting DNN models have exhibited promising performance. A more detailed explanation of graph representation in traffic prediction can be found in [29].

2.2 Representing Temporal Dependencies

Even though there are various methods for representing spatial dependencies, there are only *two* input data representation types for temporal dependencies: *sequentiality* and *periodicity*. In the former, as in the case of the stacked-vector method for spatial dependencies, consecutive temporal units are stacked into a single vector to indicate that temporally closer data are related more. This is the most common method used in almost all the studies reviewed in this work. For instance, in the temporal sequence $T = \{t_1, , t_m\}$ (t_i indicates a temporal unit), the final representation is $[V(t_1), V(t_2), , V(t_m)]$, where $V(t_i)$ is the traffic value of t_i at a specific spatial unit. The size (length) of a temporal sequence can vary depending on the task.

In the latter method, temporal periodicity (e.g., hourly, daily, weekly, or monthly patterns) is represented. This method was first introduced in [30], where two types of temporal periodicity are considered, namely, daily and weekly patterns, in addition to recent temporal sequences, that is sequentiality. For each periodicity dimension, temporal units are stacked consecutively. The type of periodicity can vary depending on the task.

3 Deep Neural Network Techniques

The current wave of DNN evolution was initiated in 2006 by the seminal work of [31]. For the first time, it was possible to reliably train a DNN in three steps: learning a stack of Restricted Boltzmann Machines (RBMs) in a layer-by-layer manner, generating a deep autoencoder with the learned layers, and finally fine-tuning through backpropagation of error derivatives. Closely following [31], RBMs and Deep Belief Networks (DBNs) based on stacked RBMs dominated the early applications of modern deep learning.

Once it became clear that training DNNs is possible, simpler training methods were investigated. Thorough studies were conducted on weight value initialization, optimization techniques, cost function design, and activation function design. Eventually, training DNNs
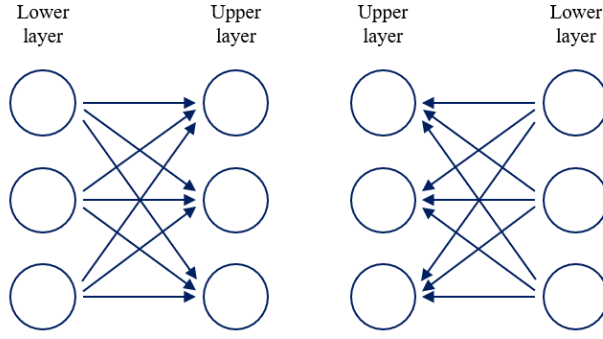
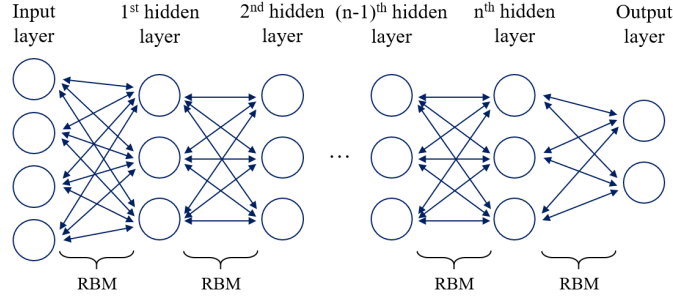Figure 3: Schematic of Restricted Boltzmann Machine (RBM).

Figure 4: Schematic of Deep Belief Network (DBN).

for well-known tasks became straightforward even for very deep networks [31]. Multi-Layer Perceptron (MLP) [32] became the default DNN architecture, whereas Convolutional Neural Networks (CNNs) [33, 34] and Recurrent Neural Networks (RNNs) [35] became the method of choice for efficiently handling image (or image-like) and sequential datasets, respectively.

With the success of MLP, CNN, and RNN, sophisticated DNN architectures and techniques were developed for various types of learning and prediction problems. Even though there are numerous such examples, we focus on the most important methods for handling STTP tasks. In the following, we categorize the DNN techniques into five different generations roughly in the order of when they first became popular. Even though such a categorization and the term *generation* are not commonly used, we include them for the purpose of clearly describing the research trends in the past decade.

3.1   Restricted Boltzmann Machines and Deep Belief Networks (First Generation)

As the first successful DNN in [31] was based on stacked RBMs trained with an unsupervised autoencoder technique, several early STTP methods were also based on the same approach.

3.1.1   Restricted Boltzmann Machines

Boltzmann Machines (BMs) originate from statistical physics. A Boltzmann machine is a graphical model with nodes and links, where each node represents a random variable, and each link determines the interaction strength between the connected nodes. A Restricted Boltzmann Machine (RBM) is a special type of BM, where the nodes in a layer can have their links connected to the nodes in the other layer only. Thereby, update algorithms can iterate over the input and the hidden layer. An example of an RBM is shown in Fig. 3; it can be viewed as a simple neural network as well.

3.1.2   Deep Belief Networks

A Deep Belief Network (DBN) is a computational model that is formed by individually learning and stacking RBMs. These networks have primarily been used as a pre-training method for supervised learning tasks; moreover, they have been used as an unsupervised feature extraction technique. Historically, DBNs are important because they were used for training early successful deep learning models. However, they gradually lost their popularity, as deep learning training was more easily performed using cross-entropy as the cost function, and Rectified Linear Unit (ReLU) as the activation function. An example of a DBN is shown in Fig. 4.

3.1.3   Autoencoder

An autoencoder is an unsupervised technique whereby an input is encoded so that it may have a small representation, and then it is decoded back to the original size. The decoded result should be as similar as possible to the original input. As shown in Fig.
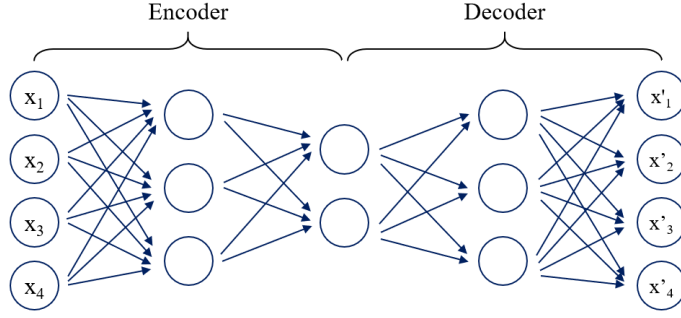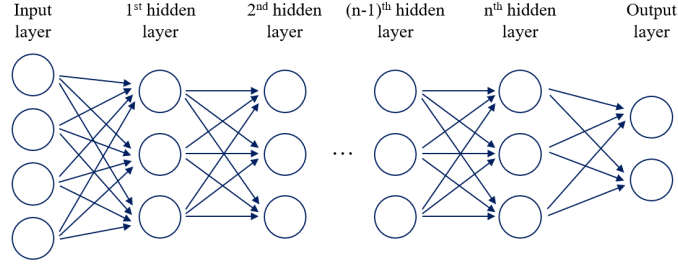
Figure 5: Schematic of autoencoder.

Figure 6: Schematic of Multi-Layer Perceptron (MLP).

5, the autoencoder architecture can be considered a feed-forward neural network with a smaller mid-layer representation and a cost function that compares the input and output. Owing to the constraints, the network is forced to learn a compact representation using a nonlinear encoder and decoder. When compared with linear methods such as principle component analysis, autoencoders can learn considerably more complicated and useful representations owing to the nonlinear nature of learning. In [36], an RBM was used for initializing each layer, and a Stacked Auto-Encoder (SAE) was used for fine-tuning the layers.

3.2 Multi-Layer Perceptron, Convolutional Neural Networks, and Recurrent Neural Networks (Second Generation)

The most fundamental and yet often sufficiently powerful DNN architectures are MLP, CNNs, and RNNs. Typical traffic data form a time-series, and sequential methods, such as the classic ARIMA and Hidden Markov Model (HMM), are popular non-DNN techniques. Among the DNN techniques, an RNN is the default neural network architecture for handling time-series data with state transition in time; therefore, RNNs have been used for numerous STTP tasks. However, it has been demonstrated that CNNs can be effective even for time-series data because capturing repeated patterns in time can be more important than modeling hidden states to improve prediction performance [37, 38].

3.2.1 Multi-Layer Perceptron

The MLP is the most basic and fundamental architecture of modern DNNs. It was originally termed the multi-layer version of the classic perceptron [39]; it is now referred to as general feedforward network with multiple hidden layers. A schematic of the MLP is shown in Fig. 6. As it is the most basic architecture, it usually serves as the baseline architecture of DNN algorithms.
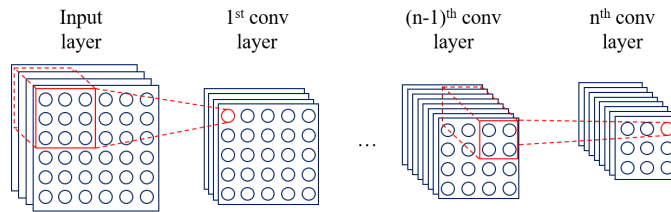

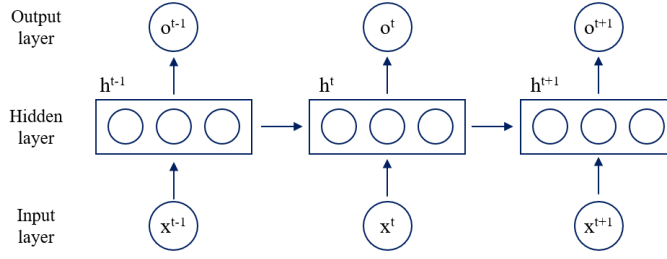Figure 7: Schematic of Convolutional Neural Network (CNN).

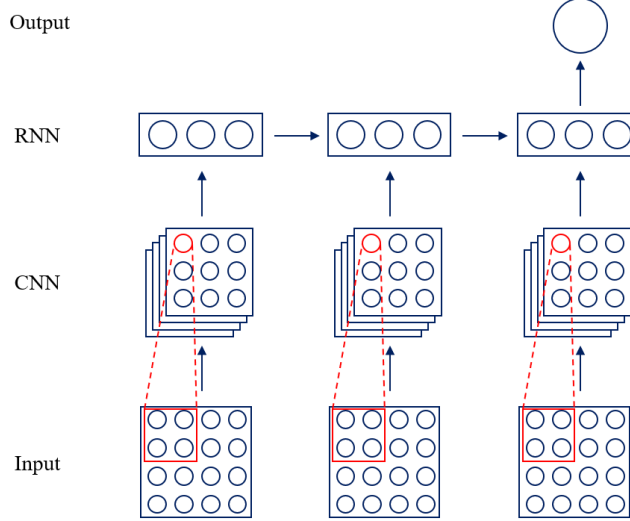Figure 8: Schematic of Recurrent Neural Network (RNN).

Figure 9: Example of hybrid model.

3.2.2 Convolutional Neural Networks

In an object detection task on an image input, a shift in the object location should not affect the output of the network. Such behavior is known as shift invariance or equivariance, and CNNs are a specialized type of neural networks designed to ensure shift invariance. In addition to this property, CNNs typically require less computation power than the MLP for a given task because they share parameters and have thus sparse interactions [40, Chapter 9]. Owing to their shift invariance, CNNs have been widely used to handle image, video, and grid-like datasets. Currently, a CNN can also be used as a general building block because it is computationally efficient and provides a satisfactory performance in a variety of tasks. In some studies on STTP, CNNs have been adopted to specifically handle images or grid parts of the data [41–43]. In several other studies, however, CNNs have been used as general building blocks. An example of a CNN is shown in Fig. 7.

3.2.3 Recurrent Neural Networks

Classic machine learning algorithms for handling time-sequential data include ARIMA and HMM. The latter is more advanced in that it can model latent states [44]. A considerably upgraded version of HMM is an RNN, which is far more sophisticated in modeling time dependencies than a Markov process, and can handle long-term dependencies for variable input and output size. For some applications such as machine translation, RNNs accumulate information over a long time interval and discard this information after using it. This mechanism was implemented using the concept of gating, and Long Short-Term Memory (LSTM) emerged as the most popular RNN model [45]. An example of an RNN is shown in Fig. 8.

Datasets for STTP usually contain spatiotemporal information with temporal information in a time-sequential form. Thus RNNs, including LSTM, have been used in numerous cases, and improvement over ARIMA, historical average, and other classic machine learning models has been reported.

3.3 Hybrids of CNN and RNN (Third Generation)

In mainstream deep learning research, the default choice of architecture has been CNN for image data and RNN for sequential data. For instance, in [46], an input image is processed by a vision processing CNN and subsequently passed through a text-generating RNN. Overall, the network generates a text description of the input image. Short-term traffic predictions usually involve spatiotemporal data, where spatial information can be mapped into image or image-like format, and temporal information can be mapped into sequential

format. Therefore, it is natural to use a hybrid network, where a CNN is applied to the spatial part, and an RNN to the temporal part. In practical implementations, however, a strong mapping between data and architecture types is not necessary because both the CNN and RNN can be used for general modeling of any data type. Mapping is recommended but is not necessarily the best-performing model. Fig. 9 shows an example of hybrid model.

3.4 Graph Convolutional Networks (Fourth Generation)

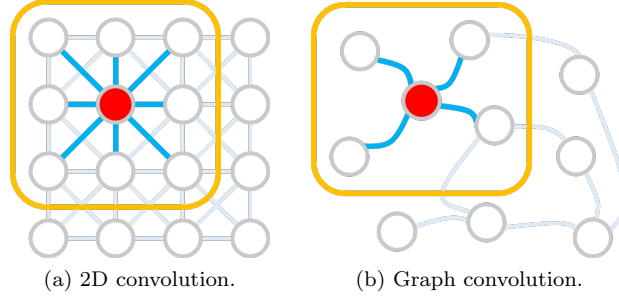(a) 2D convolution. (b) Graph convolution.

Figure 10: Typical 2D convolution and a graph convolution. Best viewed in color. (a) 2D convolution. (b) Graph convolution. (Figures were adapted from Fig. 1 of [47] and presented in Fig. 3 of [28].)

As explained in Section 2, a graph is a powerful means of representing spatial information, such as on-path proximity, k-hop connectivity, and driving directions. In a typical graph- and DNN-based modeling process, spatial relationships that can be best represented in a graph are first identified. Then, the pair-wise information among the N nodes is summarized into an $N \times N$ matrix for each graph relationship. A matrix that represents spatial interdependencies is called adjacency matrix, and its (i, j) element represents the pairwise relationship between node i and node j. For instance, the most widely used adjacency matrix is the distance matrix, where its (i, j) element contains the on-path distance information between the road segment i and road segment j. A graphical explanation is shown in Fig. 10. Once the adjacency matrices are generated, they are used as the design elements of a Graph Convolutional Network (GCN). Even though early GCNs were introduced to bridge spectral graph theory and DNNs [48], they have evolved into simpler architectures with high computational efficiency. These networks as DNN models, however, have a sufficiently large capacity and the potential for capturing complex spatial relationships despite the simplification.

Graph convolutional networks need not explicitly use convolutional layers; however, the term *convolutional* is used because of the manner in which information is packed into adjacency matrices. In Fig. 10, a typical 2D convolution is shown together with a graph convolution. In both, the information of neighboring nodes, that is, the nodes inside the filter (orange box), is used for processing the information of the node of interest (shown in red). In 2D convolution, defining the filter shape and size is straightforward because the underlying data are well structured, where all the neighboring nodes connected with blue lines are located inside the filter. However, this is not case for graph convolution. Unlike grid data, traffic network data are not well structured, and the size of the neighborhood of a node can vary. Instead of defining a small and common filter for all nodes, an $N \times N$ matrix is used as a spatial filter, where neighborhood information is collectively represented in the matrix. In row i of the matrix, all the neighboring nodes connected with blue lines from node i are represented with non-zero weight values, whereas the others are marked with zero.

3.5 Other Advanced Techniques (Fifth Generation)

The most fundamental deep learning architectures are MLP, CNN, and RNN, and a variety of techniques can be combined to augment their capabilities, as for example GCNs. We now discuss widely used techniques in STTP, as well as a few other techniques that have been considered in deep learning studies, but have not been heavily applied in transportation research.

3.5.1 Transfer Learning

Transfer learning is a machine learning technique in which knowledge is obtained by solving a problem, and the acquired knowledge is used for solving a related but different problem. For instance, we can consider learning a traffic prediction function using the data of a metropolitan city A, and then apply the learnt prediction function to another metropolitan city B. According to [49], transfer learning can be formally defined as follows:

(Transfer learning) Given a source domain D_S and learning task T_S, and a target domain D_T and learning task T_T, transfer learning is aimed at improving the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$ or $T_S \neq T_T$.

There are a variety of scenarios and methods for transfer learning, and a survey can be found in [49]. For deep learning, the most typical scenario is when a sufficient amount of labeled data is available for the source problem but not for the target problem. In this case, fine-tuning can be applied, in which a network is trained by solving the source problem. Subsequently, the network weights are updated using the data and task in target domain. Considering that several factors of traffic environments, such as traffic lights, frequent road topologies, ramps to expressways or highways, and commute time, are common, transferring knowledge from one traffic dataset to another can be an effective approach. The transfer of knowledge can be applied in the space and the time dimension (or both).
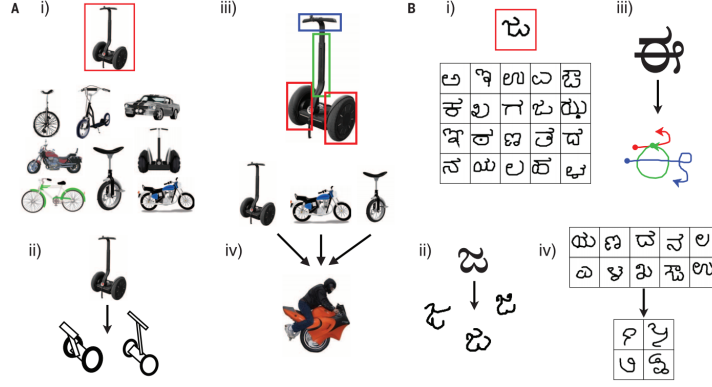
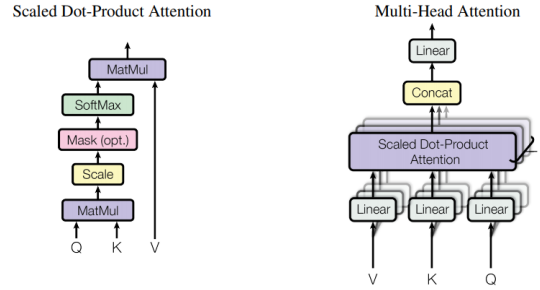Figure 11: Rich concepts can be learnt from limited data (Figure from [50]).

Figure 12: Scaled dot-product attention and multi-head attention (Figure from [55]).

3.5.2 Meta Learning

"Fast mapping" (i.e., the meaningful generalizations that can be made from one or a few examples of a novel word) is arguably the most remarkable feat of human word learning [51], and it has become popular in machine learning by a few recent studies, including [50]. Fig. 11 is provided in [50] and shows human capabilities that do not require large data. We first consider the example of a personal transporter on the left side A of Fig. 11. Even at first sight, humans can deduce that (i) this object resembles a bicycle, motorcycle, vehicle, etc. through its appearance, (ii) it can be expressed in many different forms without even knowing what it is, (iii) its components can be separated by handles, wheels, supports, etc., and (iv) it can be integrated with similar examples to generate the concept of a new transporter. On the right side B of Fig. 11, a similar example is shown for Omniglot [52], a data set of 1623 handwritten characters from 50 writing systems. In deep learning, numerous methods have been developed since 2015, where previously learnt knowledge is used for successfully completing a new but related task using only a small number of examples. This is termed meta learning and can be considered a type of transfer learning.

3.5.3 Reinforcement Learning

Reinforcement learning is fundamentally different from typical supervised or unsupervised learning in that learning is performed by interacting with the environment. More precisely, it focuses on goal-directed learning, where a mapping between the situation and the action should be learned through the interactions between the agent and the environment. Even though reinforcement learning is relatively old, it has become highly popular since the historical win over human experts in the game of "GO" [53] [54]. By integrating DNNs into reinforcement learning, the modeling accuracy of the policy and the value function was dramatically improved, where policy is defined as the mode of behavior of the learning agent at a given time, and value is defined as the total amount of rewards that an agent can expect to accumulate in the long run. In transportation, reinforcement learning can be effective if action and reward can be clearly defined. For instance, it can be useful for bike repositioning, drone path control, or autonomous driving policy control as well as properly defined STTP problems.

3.5.4 Attention

The attention mechanism is based on a simple heuristic: Humans attend to a specific part when processing a large amount of information, such as text of an image. Even though its first appearance in deep learning was primarily as a component for machine translation [56], it has now become the most important standard part in state-of-the-art Natural Language Processing (NLP) models. The Transformer architecture [55] is a DNN model that is solely based on the attention mechanism, and the technique of bidirectional encoder representations from transformers [57] is an improved transformer architecture with state-of-the-art performance in a wide range of NLP tasks. Inspired by this remarkable performance, researchers are attempting to use attention in other application areas as well. Fig. 12, which is from [55], shows the mechanism of self-attention.

3.5.5 Other Possibilities

Deep neural networks are unique in that they have millions of parameters enabling complex function modeling. They use the simple but highly effective training method of stochastic gradient descent, whereby important features and patterns may be automatically discovered. Great progress has been made in the past ten years, and the intensity of deep learning research is expected to remain high in the foreseeable future. Several new techniques useful for STTP are expected to be developed. Among the currently available techniques, generative models, such as variational autoencoders [58, 59] and generative adversarial networks [60], may be directly applicable in STTP. In addition, remarkable progress has recently been made in self-supervised learning [61, 62], which is expected to be quite useful in STTP problems, where a large amount of unlabeled data is available.

4 Application Areas

A summary of the surveyed papers are shown in Table 2 where the papers are organized in chronological order considering the DNN technique generation and publication year. However, we provide a summary of the selected papers according to application area below. We classified the application areas depending on the entity type and target measure.

4.1 Crowd Flow Prediction

Crowd flow refers to the number of people in a pre-defined region in a given time period. As there are a variety of transportation types, such as bikes and subways, or simply walking, there are no clear spatial restrictions on the moving paths. Accordingly, in most studies [30, 63–72], the regions are defined first, and then grid representation is used for spatial dependencies. In some cases, crowd flow can be divided into inflow and outflow.

In most surveyed crowd flow prediction studies, the spatial dependency is considered through grid representation based on coordinates and flow direction (i.e., inflow/outflow), and the temporal dependency is considered through sequentiality and periodicity (daily, weekly). Reference [30] was the first study to use this type of representation for crowd flow prediction. Specifically, input data are represented as 2-channel and 2-dimensional images including three spatial dependencies: Each channel corresponds to inflow and outflow and each dimension corresponds to latitude and longitude. The divided region refers to a pixel in the image, and each pixel has a crowd flow value per timestamp. In addition, to consider the sequentiality and two periodicities, three individual networks are constructed for each time period, and a fusion network is used to combine them. Later, several studies improved [30] in various ways. For example, in [70], the network from a CNN is transformed into a hybrid network (ConvGRU), and in [64], Gaussian noise is introduced into the hidden units.

In [19], the spatial dependency is represented as a multi-graph based on three types of non-Euclidean spatial dependencies of the bike dataset. For all pairs of bike stations, the distance graph is defined as the symmetrical distance, the interaction graph is defined as the number of ride records, and the correlation graph is defined as the Pearson correlation coefficient of past records. Each graph is separately learnt by an individual CNN. Then, an encoderdecoder LSTM is constructed to consider temporal correlation.

Reference [64] is aimed at effectively transferring knowledge from a data-rich source city to a data-scarce target city. The matching function is defined based on history and auxiliary data, and then the prediction network referenced from previous works, such as ST-ResNet [30], is fine-tuned.

4.2 Crowd Demand Prediction (Taxi, Bike, Ridesharing)

Crowd demand refers to the number of people with pickup or dropoff demands, such as taxi, bike-sharing, or ridesharing, in a pre-defined region in a given time period. Accurate demand prediction can lead to an efficient disposition of supplies. As in the case of crowd flow, the entity type of which is humans, there is no spatial restriction; thus, in most studies [73–77, 77–82], the regions are defined first, and then grid representation is used for the spatial dependency.

In most surveyed crowd demand prediction studies, the spatial dependency is considered through grid representation based on coordinates or graphs with non-Euclidean dependencies, and the temporal dependency is considered through sequentiality only. In [77] a hybrid network was constructed to simultaneously consider three different perspectives: a CNN for the spatial perspective, LSTM for the temporal perspective, and dynamic time warping with the embedding vectors resulting from the large-scale information network embedding algorithm for the semantic perspective.

In [18], five types of non-Euclidean spatial dependencies of the bike demand dataset were compared: spatial distance, demand matrix, average trip duration as binary, demand correlation, and fully trainable matrix. The fully trainable matrix exhibited the best performance. In [83], a multi-graph including three types of non-Euclidean spatial dependencies is defined. By representing regions as grids, the neighborhood of a region is defined based on the spatial proximity in binary, functional similarity is defined as the similarity between point-of-interest vectors, and transportation connectivity is defined based on the connectedness through motorways, highways, or public transportation, such as subway systems. For each graph, graph convolution is used individually, and then the outputs are fused. To model temporal correlation, a contextual gated recurrent neural network was further proposed. This network augments an RNN with a contextual-aware gating mechanism to re-weight different historical observations.

Table 2: Summary of all surveyed papers. Regarding **DNN Technique**, **Generation** follows the definition in Section 3, and **Model** presents only the abbreviations as follows: *TL* for transfer learning, *ML* for meta learning, *Att* for the attention network, *CN* for the capsule network, and others, as mentioned in Section 3. Regarding **Application Area**, **Entity Type** is divided into human (*H*), and vehicle (*V*), and **Target Measure** is divided into flow (*F*), speed (*S*), demand (*D*), congestion (*C*), and accidents (*A*). Regarding **Dataset**, **Availability** is confirmed at the time of writing, and **Name** follows Table 3 in Section 5. As in the case of dataset availability, **Code Availabilty** is confirmed if the code is published in the paper in question and is available at the time of writing. Regarding **Dependencies**, the abbreviations follow Table 1.

Reference	Year	DNN Technique		Application Area		Dataset		Code Avail-ability	Dependencies	
		Gene-ration	Model	Entity Type	Target Measure	Available	Name		Spatial	Temporal
Lv et al. [84]	2014	1	SAE	V	F	O	PeMS	-	1-SR	1-S
Baek et al. [85]	2016	1	DBN	H	F	-	-	-	-	-
Duan et al. [86]	2016	1	SAE	V	F	O	PeMS	-	-	1-S
Koesdwiady et al. [87]	2016	1	DBN	V	F	O	PeMS	-	-	1-S
Yang et al. [88]	2016	1	SAE	V	F	-	-	-	-	1-S
Jia et al. [89]	2016	1	DBN	V	S	-	-	-	-	1-S
Liu et al. [14]	2017	1	SAE	H	F	-	-	-	-	-
Jia et al. [90]	2017	1	DBN	V	F	-	-	-	-	1-S
Pamula et al. [91]	2018	1	SAE	V	F	-	-	-	1-SR	1-S
Zhang et al. [92]	2018	1	DBN	V	F	O	PeMS	-	1-SR	1-S
Tian et al. [93]	2015	2	RNN	V	F	O	PeMS	-	-	1-S
Ma et al. [94]	2015	2	RNN	V	S	-	-	-	-	1-S
Fu et al. [95]	2016	2	RNN	V	F	O	PeMS	-	-	1-S
Wang et al. [41]	2016	2	CNN	V	S	-	-	-	1-SF	1-S
Elhenawy et al. [96]	2016	2	MLP	V	S,F	-	-	-	1-SR	1-S
Wang et al. [97]	2017	2	MLP	H	D	-	-	-	-	-
Sudo et al. [98]	2017	2	MLP	H	F	-	-	-	-	1-S
Zhang et al. [30]	2017	2	CNN	H	F	O	TaxiBJ,BikeNYC	Keras	3-GIO	3-SWD
Huang et al. [73]	2017	2	MLP	H,V	D,C	-	-	-	2-G	-
Yuan et al. [99]	2017	2	MLP	V	A	-	-	-	-	-
Dai et al. [100]	2017	2	RNN	V	F	O	PeMS	-	-	1-S
Kang et al. [101]	2017	2	RNN	V	F	O	PeMS	-	-	1-S
Zhao et al. [102]	2017	2	RNN	V	F	-	-	-	1-SR	1-S
Jia et al. [103]	2017	2	RNN	V	S	-	-	-	-	1-S
Ma et al. [42]	2017	2	CNN	V	S	-	-	-	1-SR	-
Yu et al. [16]	2017	2	RNN,SAE	V	S	O	PeMS	-	-	1-S
Polson et al. [104]	2017	2	MLP	V	S,F	O	I55	-	1-SR	1-S
Sun et al. [105]	2017	2	CNN	V	C	O	HERE API	-	2-G	1-S
Xu et al. [106]	2018	2	RNN	H	D	-	-	-	-	1-S
Chen et al. [63]	2018	2	CNN	H	F	O	TaxiBJ,BikeNYC	-	3-GIO	3-SW
Wang et al. [64]	2018	2	CNN	H	F	O	TaxiBJ,BikeNYC	-	3-GIO	2-SW
Zhang et al. [65]	2018	2	CNN	H	F	O	TaxiBJ,BikeNYC	Keras	3-GIO	3-SWD
Abbas et al. [107]	2018	2	RNN	V	F	-	-	-	1-SR	1-S
Xie et al. [108]	2018	2	RNN	V	F	-	-	-	-	1-S
Zhang et al. [109]	2018	2	RNN	V	F	O	PeMS	-	-	1-S
Zhao et al. [110]	2018	2	RNN	V	F	O	PeMS	-	-	1-S
Adu-Gyamfi et al. [111]	2018	2	RNN	V	S	-	-	-	-	1-S
Cui et al. [112]	2018	2	RNN	V	S	O	STAR	-	1-SR	1-S
Fandango et al. [113]	2018	2	RNN	V	S	O	PeMS	-	-	1-S
Liao et al. [114]	2018	2	RNN	V	S	-	-	-	1-SR	1-S
Shen et al. [43]	2018	2	CNN	V	S	-	-	-	1-SF	1-S
Lin et al. [66]	2019	2	CNN	H	F	O	MobileBJ,BikeNYC	-	3-GIO	3-SWD
Rong et al. [67]	2019	2	CNN	H	F	-	-	-	3-GIO	1-S
Zhang et al. [68]	2019	2	CNN	H	F	O	TaxiBJ,BikeNYC	-	3-GIO	3-SWD
Xiangxue et al. [115]	2019	2	RNN	V	F	-	-	-	1-Gr	1-S
Sun et al. [116]	2019	2	CNN,RNN	V	S,C	-	-	-	1-SR	1-S
Zhao et al. [23]	2019	2	RNN	V	S	-	-	-	-	1-S
Wu et al. [117]	2016	3	Hybrid	V	F	O	PeMS	-	1-SR	3-SWD
Ke et al. [74]	2017	3	Hybrid	H	D	-	-	-	2-G	1-S
Du et al. [118]	2017	3	Hybrid	V	F	O	PeMS	-	-	1-S
Liu et al. [119]	2017	3	Hybrid	V	F	O	PeMS	-	1-SR	3-SWD
Yu et al. [120]	2017	3	Hybrid	V	S	-	-	-	2-G	1-S
Liao et al. [75]	2018	3	Hybrid	H	D	O	TaxiNYC	-	2-G	3-SWD
Wang et al. [76]	2018	3	Hybrid	H	D	O	Didi	-	2-G	1-S
Yao et al. [77]	2018	3	Hybrid	H	D	X	-	Keras	2-G	1-S
Ma et al. [69]	2018	3	Hybrid	H	F	-	-	-	2-G	1-S
Zonoozi et al. [70]	2018	3	Hybrid	H	F	O	TaxiBJ	-	3-GIO	3-SWD
Yang et al. [121]	2018	3	Hybrid	V	F	-	-	-	1-SR	1-S
Duan et al. [122]	2018	3	Hybrid	V	F	-	-	-	3-GIO	1-S
Ren et al. [71]	2019	3	Hybrid	H	F	-	-	-	3-GIO	3-SWD
Li et al. [15]	2017	4	Graph	V	S	O	PeMS	TF, Pytorch	1-Gr	1-S
Yu et al. [123]	2017	4	Graph	V	S	O	PeMS	TF	1-Gr	1-S
Chu et al. [17]	2018	4	Graph	H	D	X	-	-	1-Gr	2-SD
Lin et al. [18]	2018	4	Graph	H	D	O	BikeNYC	-	1-Gr	1-S
Chai et al. [19]	2018	4	Graph	H	F	O	BikeNYC,BikeCHI	-	3-Gr	1-S
Iyer et al. [21]	2018	4	Graph	V	S	O	BusNYC	-	1-Gr	1-S
Lv et al. [22]	2018	4	Graph	V	S	-	-	-	1-Gr	3-SWD
Zhao et al. [110]	2018	4	Graph	V	S	O	SZ-Taxi,PeMS	TF	1-Gr	1-S
Zhang et al. [24]	2019	4	Graph	V	S	-	-	-	1-Gr	1-S
Cui et al. [20]	2019	4	Graph	V	S	-	-	-	1-Gr	1-S

(Table 2 Continued from the previous page)

| Reference | Year | DNN Technique | | Application Area | | Dataset | | Code Avail-ability | Dependencies | |
		Gene-ration	Model	Entity Type	Target Measure	Available	Name		Spatial	Temporal
Zhang et al. [25]	2019	4	Graph	V	S,F	O	PeMS	-	1-Gr	3-SWD
Du et al. [26]	2020	4	Graph	H	D	O	TaxiNYC,BikeNYC	-	1-Gr	1-S
Lv et al. [27]	2020	4	Graph,RNN	V	F	O	PeMS	-	3-Gr	1-S
Lee et al. [28]	2020	4	Graph	V	S	O	TOPIS	-	3-Gr	1-S
Cheng et al. [124]	2018	5	Att	V	F	O	MapBJ	-	1-SF	1-S
Ning et al. [125]	2018	5	RL	H	D	-	-	-	-	1-S
Rodrigues et al. [126]	2018	5	Att	H	D	O	TaxiNYC	-	-	1-S
Zhou et al. [78]	2018	5	Att	H	D	O	TaxiNYC,BikeNYC	-	2-G	1-S
Li et al. [127]	2018	5	RL	H	D	O	BikeNYC	-	1-Gr	1-S
Wang et al. [128]	2018	5	TL	H	F	O	TaxiBJ, BikeNYC	-	2-G	1-S
Yao et al. [129]	2018	5	Graph,Att	H,V	D,F	O	TaxiNYC,BikeNYC	Keras	2-G,1-Gr	1-S
Du et al. [130]	2018	5	Att	V	F	O	HighwaysUK	-	-	1-S
Liang et al. [131]	2018	5	GAN	V	F	O	STAR	-	1-SR	1-S
Wu et al. [132]	2018	5	Graph,Att	V	F	X	-	-	1-Gr	1-S
Wu et al. [133]	2018	5	Att	V	F	O	PeMS	-	1-SR	3-SWD
Liu et al. [134]	2018	5	Att	V	S	O	PeMS	-	1-SR	1-S
Ma et al. [135]	2018	5	CN	V	S	-	-	-	2-G	1-S
He et al. [136]	2018	5	Att	V	S	O	HK	-	1-SR	1-S
Geng et al. [83]	2019	5	Graph,Att	H	D	-	-	-	3-Gr	1-S
Yao et al. [79]	2019	5	ML,Att	H	D	O	TaxiNYC,Porto, BikeNYC,BikeDC, BikeCHI	-	2-G	1-S
He et al. [80]	2019	5	CN,RL	H	D	O	TLC,Uber,Didi	-	2-G	-
Mourad et al. [137]	2019	5	Att	H	F	O	BikeNYC	TF	3-GIO	1-S
Zhou et al. [138]	2019	5	Att	H	F	O	TaxiBJ, TaxiNYC,BikeNYC	Keras	2-G	1-S
Zhang et al. [139]	2019	5	TL	V	A	O	Twitter,NYPD	-	-	-
Guo et al. [140]	2019	5	Graph,Att	V	F	O	PeMS	MXNet	1-Gr	3-SWD
Pan et al. [141]	2019	5	Graph,Att,ML	V	S,F	O	T-Drive,PeMS	MXNet	1-Gr	1-S
Zheng et al. [142]	2019	5	Graph,Att	V	S	O	PeMS,Xiamen	-	1-Gr	1-S
Zhang et al. [143]	2019	5	Graph,Att,RNN	V	S	O	PeMS	-	1-Gr	1-S
Zhang et al. [144]	2019	5	GAN	V	S,F	O	ChicagoBus	-	1-SR	3-SWD
Yi et al. [145]	2019	5	Att	V	S,F	-	-	-	1-SR	3-SWD
Li et al. [72]	2019	5	Att,CNN,RNN	H	F	O	BikeNYC,TaxiBJ	Keras	3-GIO	1-S
Zhou et al. [81]	2020	5	Att	H	D	O	TaxiBJ, TaxiNYC,BikeNYC	Keras	2-G	1-S
Cai et al. [146]	2020	5	Graph,Att	V	S	O	PeMS	-	1-Gr	3-SWD

4.3 Traffic Flow Prediction

Traffic flow refers to the number of vehicles passing through a spatial unit, such as a road segment or traffic sensor point, in a given time period. Accurate prediction assists in preventing traffic congestion and managing traffic conditions in advance. Unlike humans, vehicles can move only through pre-defined roads; accordingly, stacked vector and graphs are often used for the input data representation.

In one of the first STTP studies, SAE networks were trained [84]. Stacked vector and sequentiality were used for the input data representation. In [25], the spatial dependency is represented as a graph based on the temporal correlation coefficient using historical traffic observations to capture heterogeneous spatial correlations. To calculate the correlation coefficient, min-max normalization is first used to calculate the coefficient so that spatial heterogeneity may be eliminated from capacity or speed limits. Then, the daily periodicity is also removed using the z-score transformation because it may result in strong temporal auto-correlation. Finally, the network is trained including graph convolution and residual LSTM.

In [141], ST-MetaNet was proposed. Its input data are represented as graphs based on connecting patterns or distance. A sequence-to-sequence architecture consisting of an encoder for learning historical information, and a decoder for step-by-step prediction are employed. Specifically, the encoder and decoder have the same network structure: a meta graph attention network to capture spatial correlations, and a meta recurrent neural network to consider temporal correlations.

4.4 Traffic Speed Prediction

Traffic speed refers to the average speed of vehicles passing through a spatial unit, such as a road segment or traffic sensor point, in a given time period. As in the case of traffic flow, the entity type of which is vehicles, stacked vectors and graphs are often used for the input data representation.

Traffic speed prediction can be divided into two types according to the traffic network considered. The first type considers only freeways. As freeways have a relatively simple structure with a few traffic signals or on/off-ramps, prediction accuracy can be improved by simply increasing the temporal resolution of the dataset. Hence, in the case of traffic speed prediction for freeways, temporal dependency has been emphasized more than spatial dependency. Therefore, input data are often represented by simple stacked-vector methods. In [42], the input data are represented as a 2-d matrix combining the spatial and temporal dependencies. Each column of the matrix corresponds to the stacked vector in the connection order, and each row represents the temporal sequentiality.

In recent studies, more complex traffic networks, such as urban networks, have been considered. These networks may have considerably more complicated connection patterns and abrupt changes. Thus, more sophisticated spatial dependency representation methods, such as graphs considering proximity or pairwise distance, have been introduced. In [15], bidirectional diffusion convolution is performed on the graph to capture spatial dependency, and a sequence-to-sequence architecture with gated recurrent units is used to capture temporal dependency. Input graphs are defined based on the shortest path distance between traffic sensors. Later, in [16], the training time was shortened by replacing recurrent layers by convolutional layers. Recently, in [28], the input data are represented as multi-graphs based on three types of spatial dependencies: distance, direction, and positional relationships. The graph weights are also modified using simple partition filters.

Reference [135] proposed a network consisting of a capsule network with grid representation for the input data and nested LSTM to consider temporal sequentiality. In [24], a distance-based graph is first transformed into vectors. Then, a spatial embedding is trained to encode vertices into vectors that preserve the graph structure information. The attention network is manipulated for the final prediction.

4.5 Traffic accidents/congestion prediction

Unlike the aforementioned applications corresponding to the regression problem, traffic accident/congestion prediction is related to the classification problem. For accident prediction, some studies predict whether an accident occurs through a binary classification, and others predict the injury levels resulting from accidents through a multi-class classification. For congestion prediction, the congestion level is determined by the specific traffic flow or traffic speed value. Typical studies predict congestion through a binary classification; congestion is defined based on an average speed less than 20 km/h.

Reference [139] proposed transferring the prediction network from the source area, which has the ground-truth traffic accident reports, to the target area, which does not. Traffic-accident-related features, such as accident location and time, are first extracted from unstructured social sensing data (e.g., tweets). Then, a transformation network and a loss function are defined to consider the discrepancy between the source and target distributions. Finally, all networks are trained using the adversarial method.

In [116], the congestion level is predicted from a CNN and an RNN, which are individually trained. The input data for the CNN are represented by stacked vectors and temporal sequentiality, whereas the input data for the RNN are represented by the temporal sequentiality of the average speed over multiple spatial units. Finally, the traffic congestion levels are classified according to the output of the CNN and RNN.

5 Public Datasets

A major problem in STTP studies is the lack of standard benchmark datasets [1]. Benchmark datasets can greatly facilitate research because findings can be easily compared and integrated, whereas the lack of benchmark datasets may result in data-specific rather than generalizable studies. Accordingly, we summarize publicly available often-used datasets in Table 3. Dataset availability was determined at the time of writing.

Table 3: Summary of Public Datasets. It should be noted that dataset names are not official.

Dataset	City, Country	Duration	Time Resolution	Spatial Coverage	Data Type (Examples)
PeMS [147]	California, USA	2001∼	5min	Nearly 40,000 individual detectors, spanning the freeway system across all major metropolitan areas	Traffic detectors, incidents, lane closures, toll tags, census traffic counts, vehicle classification, weight-in-motion, roadway inventory, etc.
QTraffic [148]	Beijing, China	4/1 2017 ∼5/31 2017	15min	15,073 road segments covering approximately 738.91 km (6th ring road)	Traffic speed, user queries (timestamp, starting/destination location), Road network information (width, direction, length, etc.)
TaxiNYC [149]	New York, USA	2009∼	- (Per trip)	Including Bronx, Brooklyn, Manhattan, Queens, Staten Island	Pick-up and drop-off date/time/location, trip distance, rate/payment type, passenger count
TaxiBJ [150]	Beijing, China	7/1 2013 ∼4/30 2016	30 min	Over 34,000 taxis	Trajectory, meteorological data
BikeNYC [151]	New York, USA	5/27 2013∼	- (Per trip)	Over 6,800 bikes	Trip duration, start/end station ID/time, bike ID, user type, gender, year of birth
BusNYC [152]	New York, USA	2014	10min	All public buses	Coordinates (latitude, longitude), timestamp of transmission, distance to the next bus stop
WebTRIS [153]	England	4.2015∼	15min	All motorways and 'A' roads managed by the Highways England (Strategic Road Network)	Journey time, traffic flow, longitude/latitude/length of link, flow quality, quality index
DRIVENet [154]	Seattle, USA	2011	1min	323 stations, 85 miles	Pedestrian travels, public transit data, traffic flow, travel time, safety performance, incident induced delay
Porto [155]	Porto, Portugal	7/1 2013 ∼6/30 2014	- (Per trip)	442 taxis, 420,000 trajectories, 16735m*14389m	call type, timestamp, day type, GPS coordinates
Didi [156]	Chengdu, China	11/1 2016 ∼11/30 2016	2∼4secs	5476 geographical blocks (each block size = 1 km^2)	Order ID, start/end billing time, pick-up/drop-off longitude/latitude
BikeDC [157]	Washington, USA	9/20 2010 ∼	- (Per trip)	Over 500 stations across 6 jurisdictions, 4,300 bikes	Duration, start/end date, start/end station, bike number, member type
BikeCHI [158]	Chicago, USA	6/27 2013∼	- (Per trip)	580 stations across Chicagoland, 5,800 bikes	Trip start/end time/station, rider type (member, single ride, explore pass)
HK [159]	Hong Kong, Hong Kong	12/28 2015∼	2 min	four regions in Hong Kong	Link ID, region, road type, road saturation level (good/bad/average), speed, date
Uber [160]	New York, USA	4/1 2014∼9/30 2014, 1/1 2015∼6/30 2015	- (Per trip)	Over 18.8 million pickups	Date/time, latitude, longitude
TOPIS [161]	Seoul, South Korea	2014.1∼	5 min, 1 h	1,153 sensors and over 70,000 taxis(GPS)	Date/time, coordinates, road type, region type, average speed/flow, number of lanes
BusCHI [162]	Chicago, USA	8/2 2011 ∼ 5/3 2018	10 min	Buses on arterial streets in real-time, about 1,250 road segments covering 300 miles	Time, segment ID, bus count, reading count, speed

The most widely used dataset for traffic flow and speed prediction area is the Caltrans performance measurement system, called the *PeMS* dataset (used in 27 papers in this survey). This dataset is managed by the California Department of Transportation. All data are available from 2001 to the present. The *PeMS* dataset is based on nearly 40,000 individual detectors, spanning the freeway system across all major metropolitan areas of California. The final data resolution is 5 min, aggregated from 30 s raw measurements. The *PeMS* dataset is provided as a *csv* file, including information on traffic detectors, incidents, lane closures, toll tags, census traffic counts, vehicle classification, weight-in-motion, roadway inventory, etc. This dataset includes abundant spatiotemporal information; however, it is limited to simple freeways only. Thus, the insights from the studies based on this dataset may not be directly used for prediction in complex urban networks.

For crowd flow and demand prediction, there are three often-used datasets: *BikeNYC*, *TaxiBJ*, and *TaxiNYC*. The data in *BikeNYC* (used in 18 papers in this survey) have been collected from CitiBike, New York, USA, since 2013. The data are from more than 6,800 membership users in New York City, with a time resolution of 1 h. The *BikeNYC* dataset includes various information, such as trip duration, starting/ending station, starting/ending time, bike ID, user type, gender, and year of birth. The dataset can be downloaded in *csv* format.

The data in *TaxiBJ* (used in nine papers in this survey) have been collected from Beijing, China, for over three years. These data are provided in [30]. They have a time resolution of 30 min. The dataset includes trajectory and meteorology data from over 34,000 taxis in Beijing. It is primarily used for crowd flow research.

The data in *TaxiNYC* (used in nine papers in this study) were collected from NYC Taxi and Limousine Commission by technology providers authorized under the Taxicab & Livery Passenger Enhancement Programs from January 2009 to June 2019. This dataset contains taxi trip records including pick-up/drop-off time and location, trip distance, itemized fare, rate type, payment type, and driver-reported passenger count. It is primarily used for demand prediction.

6 Discussion and Conclusion

We reviewed recent DNN-based STTP research according to input data representation methods, DNN techniques, application areas, and dataset availability. Herein, we suggest challenges and possible directions for future work.

6.1 Input Data Representation Methods

Various factors are considered to determine input data representation methods. One is the entity type of a given task. For example, as a vehicle can only move on the road, using a grid representation may be inefficient owing to sparsity and overlap of the roads. In contrast, as humans can move freely, grid representation has been used in several studies. In STTP for humans, however, the input need not be represented in grid format for all cases. For instance, there could be a stronger relationship that is irrelevant to the distance or proximity (as mentioned in Section 2). In this case, locality would not be the most important information; thus grid representation may not be effective.

Recently, graph representation has been widely adopted in several applications. Any spatial unit, such as grid and road segment, can be represented as a node in a graph. As graphs can express arbitrary relationships between nodes, they may be a possible universal representation method for traffic networks with complex latent dependencies. However, to construct a graph representation, the adjacency matrix should be first obtained using domain knowledge. Thus, the graph represents only the pre-defined spatial dependency.

Finally, input data representation methods should be determined in consideration of various factors, including available dataset types, size of available datasets, domain knowledge, task characteristics, and deep network type. Thus, an appropriate design scheme for input data representation methods, such as analyzing which factor should be prioritized, and developing a new type of representation method, is required.

6.2 DNN Techniques

In most cases, each deep network employs a specific inductive bias. Through data analysis or domain knowledge, we can guess which inductive bias should be tried and enforced for each STTP problem. For instance, if there is a large dependency between proximate regions or timestamps, and this dependency is translation invariant, locality and translation invariance should be presented through the convolutional layers. By contrast, if there is no dependency between local regions or timestamps, convolutional layers would be less likely to be helpful. The availability and quality of dataset also plays an important role on deciding the DNN techniques to try. Even when the appropriate inductive bias is obvious for the given problem, we can employ a specific type of deep network only when the necessary dataset is available such that the information is extractable with deep networks.

By the no-free-lunch theorem [163], there is no consistently best-performing network for all the tasks. Therefore, it is inevitable to try multiple DNN models and choose the one with the best validation performance. When trying multiple models, knowledge on DNN technique's inductive bias and the characteristics of task and data should be fully utilized for an efficient modeling under limited time and computing resources. Furthermore, it can be helpful or even critical to consider criteria other than performance such as the required data size and training time.

6.3 Dataset and Code Availability

As mentioned in Section 5, there is no clearly established standard datasets in the STTP area for benchmarking performance and integrating new findings. Therefore, it will be very helpful to agree on a few common datasets for each application area. Table 3 can be a good start. This is an important matter especially for deep-learning based research. In deep-learning research, it is well known that the prediction accuracy for a specific dataset or task can be very sensitive to the choice of hyperparameters. For instance, modifying network depth or width can cause the performance ranking of the studied methods to be altered or even completely reversed. A sound study requires sufficient and fair amount of effort to be applied to each method such that meaningful findings can be reported. When each study uses its own dataset without other research groups verifying the results, it becomes difficult to tell if the findings are due to hyperparameter tuning or due to the newly proposed idea. Reproducibility is also an important issue in deep-learning research. For this reason, ideally not only the dataset but also the code should be made available.

6.4 Future Mobilities

Recently, various new transportation systems, such as personal transporters (e.g., electric scooters, electric kickboards, and hoverboards), have been introduced. These vehicles are considerably smaller and more personalized, and their number is expected to continue to increase. Therefore, a proper management system as a part of the ITS should be developed to prevent accidents and congestion, and the relevant STTP algorithms should be studied. These new mobilities have no movement constraints as in the case of humans, but they are significantly faster with more complex movement patterns. As new datasets are released, appropriate studies should be conducted. Similarly, autonomous vehicles might become a crucial component of transportation networks [164], and it might lead to an active development of STTP models for a mixed autonomous/human driving environment.

Drone is another example of future mobilities. Until now, drones have been mainly used for freight transportation. In the near future, however, we can expect them to be used for passenger transportation as well. Drones have quite different characteristics compared to the traditional transportation systems. They move in airspace and can change their routes in three dimensional space. They need to consider geometry of the local obstacles such as houses and buildings. Their operation is significantly affected by the weather condition. Therefore, understanding the use of deep neural networks in the application area of drone will be an important and interesting topic.

7 Acknowledgment

References

[1] M. Veres, M. Moussa, Deep learning for intelligent transportation systems: A survey of emerging trends, IEEE Transactions on Intelligent transportation systems (2019).

[2] L. Zhu, F. R. Yu, Y. Wang, B. Ning, T. Tang, Big data analytics in intelligent transportation systems: A survey, IEEE Transactions on Intelligent Transportation Systems 20 (1) (2018) 383–398.

[3] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, C. Chen, Data-driven intelligent transportation systems: A survey, IEEE Transactions on Intelligent Transportation Systems 12 (4) (2011) 1624–1639.

[4] H. Touvron, A. Vedaldi, M. Douze, H. J'egou, Fixing the train-test resolution discrepancy: Fixefficientnet, arXiv preprint arXiv:2003.08237 (2020).

[5] Q. Xie, M.-T. Luong, E. Hovy, Q. V. Le, Self-training with noisy student improves imagenet classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10687–10698.

[6] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim, et al., English conversational telephone speech recognition by humans and machines, arXiv preprint arXiv:1703.02136 (2017).

[7] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, G. Zweig, Achieving human parity in conversational speech recognition, arXiv preprint arXiv:1610.05256 (2016).

[8] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning requires rethinking generalization, arXiv preprint arXiv:1611.03530 (2016).

[9] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al., Relational inductive biases, deep learning, and graph networks, arXiv preprint arXiv:1806.01261 (2018).

[10] W. R. Tobler, A computer movie simulating urban growth in the detroit region, Economic geography 46 (sup1) (1970) 234–240.

[11] A. Krizhevsky, V. Nair, G. Hinton, Cifar-10 (canadian institute for advanced research).
URL http://www.cs.toronto.edu/~kriz/cifar.html

[12] . Krizhevsky, Cifar100.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, in: CVPR09, 2009.

[14] L. Liu, R.-C. Chen, A novel passenger flow prediction model using deep learning methods, Transportation Research Part C: Emerging Technologies 84 (2017) 74–91.

[15] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, arXiv preprint arXiv:1707.01926 (2017).

[16] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, Y. Liu, Deep learning: A generic approach for extreme condition traffic forecasting, in: Proceedings of the 2017 SIAM international Conference on Data Mining, SIAM, 2017, pp. 777–785.

[17] J. Chu, K. Qian, X. Wang, L. Yao, F. Xiao, J. Li, X. Miao, Z. Yang, Passenger demand prediction with cellular footprints, in: 2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), IEEE, 2018, pp. 1–9.

[18] L. Lin, Z. He, S. Peeta, Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach, Transportation Research Part C: Emerging Technologies 97 (2018) 258–276.

[19] D. Chai, L. Wang, Q. Yang, Bike flow prediction with multi-graph convolutional networks, in: Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2018, pp. 397–400.

[20] Z. Cui, K. Henrickson, R. Ke, Y. Wang, Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting, IEEE Transactions on Intelligent Transportation Systems (2019).

[21] S. R. Iyer, K. Boxer, L. Subramanian, Urban traffic congestion mapping using bus mobility data., in: UMCit@ KDD, 2018, pp. 7–13.

[22] Z. Lv, J. Xu, K. Zheng, H. Yin, P. Zhao, X. Zhou, Lc-rnn: A deep learning model for traffic speed prediction., in: IJCAI, 2018, pp. 3470–3476.

[23] J. Zhao, Y. Gao, Z. Yang, J. Li, Y. Feng, Z. Qin, Z. Bai, Truck traffic speed prediction under non-recurrent congestion: Based on optimized deep learning algorithms and gps data, IEEE Access 7 (2019) 9116–9127.

[24] Z. Zhang, M. Li, X. Lin, Y. Wang, F. He, Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies, Transportation research part C: emerging technologies 105 (2019) 297–322.

[25] Y. Zhang, T. Cheng, Y. Ren, K. Xie, A novel residual graph convolution deep learning model for short-term network-based traffic forecasting, International Journal of Geographical Information Science (2019) 1–27.

[26] B. Du, X. Hu, L. Sun, J. Liu, Y. Qiao, W. Lv, Traffic demand prediction based on dynamic transition convolutional neural network, IEEE Transactions on Intelligent Transportation Systems (2020).

[27] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, S. Ji, Temporal multi-graph convolutional network for traffic flow prediction, IEEE Transactions on Intelligent Transportation Systems (2020).

[28] K. Lee, W. Rhee, Graph convolutional modules for traffic forecasting, CoRR abs/1905.12256 (2019). arXiv:1905.12256.
URL http://arxiv.org/abs/1905.12256

[29] J. Ye, J. Zhao, K. Ye, C. Xu, How to build a graph-based deep learning architecture in traffic domain: A survey, arXiv preprint arXiv:2005.11691 (2020).

[30] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[31] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 1097–1105.
URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[32] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359366.

[33] K. Fukushima, S. Miyake, Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition, in: Competition and cooperation in neural nets, Springer, 1982, pp. 267–285.

[34] Y. LeCun, P. Haffner, L. Bottou, Y. Bengio, Object recognition with gradient-based learning, in: Shape, contour and grouping in computer vision, Springer, 1999, pp. 319–345.

[35] M. I. Jordan, Serial order: A parallel distributed processing approach, in: Advances in psychology, Vol. 121, Elsevier, 1997, pp. 471–495.

[36] G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507. doi:10.1126/science.1127647.
URL http://www.ncbi.nlm.nih.gov/sites/entrez?db=pubmed&uid=16873662&cmd=showdetailview&indexed=google

[37] S. Bai, J. Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, arXiv preprint arXiv:1803.01271 (2018).

[38] G. Tang, M. Müller, A. Rios, R. Sennrich, Why self-attention? a targeted evaluation of neural machine translation architectures, arXiv preprint arXiv:1808.08946 (2018).

[39] F. Rosenblatt, Principles of neurodynamics. perceptrons and the theory of brain mechanisms, Tech. rep., Cornell Aeronautical Lab Inc Buffalo NY (1961).

[40] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep learning, Vol. 1, MIT press Cambridge, 2016.

[41] J. Wang, Q. Gu, J. Wu, G. Liu, Z. Xiong, Traffic speed prediction and congestion source exploration: A deep learning method, in: 2016 IEEE 16th International Conference on Data Mining (ICDM), IEEE, 2016, pp. 499–508.

[42] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction, Sensors 17 (4) (2017) 818.

[43] G. Shen, C. Chen, Q. Pan, S. Shen, Z. Liu, Research on traffic speed prediction by temporal clustering analysis and convolutional neural network with deformable kernels (may, 2018), IEEE Access 6 (2018) 51756–51765.

[44] L. R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286.

[45] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[46] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3156–3164.

[47] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S. Y. Philip, A comprehensive survey on graph neural networks, IEEE Transactions on Neural Networks and Learning Systems (2020).

[48] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and locally connected networks on graphs, arXiv preprint arXiv:1312.6203 (2013).

[49] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (10) (2009) 1345–1359.

[50] B. M. Lake, R. Salakhutdinov, J. B. Tenenbaum, Human-level concept learning through probabilistic program induction, Science 350 (6266) (2015) 1332–1338.

[51] S. Carey, E. Bartlett, Acquiring a single new word. (1978).

[52] B. M. Lake, R. Salakhutdinov, J. B. Tenenbaum, Human-level concept learning through probabilistic program induction, Science 350 (6266) (2015) 1332–1338.

[53] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of go with deep neural networks and tree search, nature 529 (7587) (2016) 484–489.

[54] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al., Mastering the game of go without human knowledge, nature 550 (7676) (2017) 354–359.

[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[56] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).

[57] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[58] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[59] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, L. Carin, Variational autoencoder for deep learning of images, labels and captions, in: Advances in neural information processing systems, 2016, pp. 2352–2360.

[60] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[61] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, arXiv preprint arXiv:1909.11942 (2019).

[62] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, arXiv preprint arXiv:2002.05709 (2020).

[63] C. Chen, K. Li, S. G. Teo, G. Chen, X. Zou, X. Yang, R. C. Vijay, J. Feng, Z. Zeng, Exploiting spatio-temporal correlations with multiple 3d convolutional neural networks for citywide vehicle flow prediction, in: 2018 IEEE international conference on data mining (ICDM), IEEE, 2018, pp. 893–898.

[64] B. Wang, Z. Yan, J. Lu, G. Zhang, T. Li, Explore uncertainty in residual networks for crowds flow prediction, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–7.

[65] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, T. Li, Predicting citywide crowd flows using deep spatio-temporal residual networks, Artificial Intelligence 259 (2018) 147–166.

[66] Z. Lin, J. Feng, Z. Lu, Y. Li, D. Jin, Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 1020–1027.

[67] C. Rong, J. Feng, Y. Li, Deep learning models for population flow generation from aggregated mobility data, in: Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers, 2019, pp. 1008–1013.

[68] J. Zhang, Y. Zheng, J. Sun, D. Qi, Flow prediction in spatio-temporal networks based on multitask deep learning, IEEE Transactions on Knowledge and Data Engineering (2019).

[69] X. Ma, J. Zhang, B. Du, C. Ding, L. Sun, Parallel architecture of convolutional bi-directional lstm neural networks for network-wide metro ridership prediction, IEEE Transactions on Intelligent Transportation Systems 20 (6) (2018) 2278–2288.

[70] A. Zonoozi, J.-j. Kim, X.-L. Li, G. Cong, Periodic-crn: A convolutional recurrent model for crowd density prediction with recurring periodic patterns., in: IJCAI, 2018, pp. 3732–3738.

[71] Y. Ren, H. Chen, Y. Han, T. Cheng, Y. Zhang, G. Chen, A hybrid integrated deep learning model for the prediction of citywide spatio-temporal flow volumes, International Journal of Geographical Information Science (2019) 1–22.

[72] W. Li, W. Tao, J. Qiu, X. Liu, X. Zhou, Z. Pan, Densely connected convolutional networks with attention lstm for crowd flows prediction, IEEE Access 7 (2019) 140488–140498.

[73] Z. Huang, Z. Zhao, E. Shijia, C. Yu, G. Shan, T. Li, J. Cheng, J. Sun, Y. Xiang, Prace: A taxi recommender for finding passengers with deep learning approaches, in: International Conference on Intelligent Computing, Springer, 2017, pp. 759–770.

[74] J. Ke, H. Zheng, H. Yang, X. M. Chen, Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach, Transportation Research Part C: Emerging Technologies 85 (2017) 591–608.

[75] S. Liao, L. Zhou, X. Di, B. Yuan, J. Xiong, Large-scale short-term urban taxi demand forecasting using deep learning, in: 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC), IEEE, 2018, pp. 428–433.

[76] D. Wang, Y. Yang, S. Ning, Deepstcl: A deep spatio-temporal convlstm for travel demand prediction, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.

[77] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, Z. Li, Deep multi-view spatial-temporal network for taxi demand prediction, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[78] X. Zhou, Y. Shen, Y. Zhu, L. Huang, Predicting multi-step citywide passenger demands using attention-based neural networks, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 736–744.

[79] H. Yao, Y. Liu, Y. Wei, X. Tang, Z. Li, Learning from multiple cities: A meta-learning approach for spatial-temporal prediction, in: The World Wide Web Conference, 2019, pp. 2181–2191.

[80] S. He, K. G. Shin, Spatio-temporal capsule-based reinforcement learning for mobility-on-demand network coordination, in: The World Wide Web Conference, 2019, pp. 2806–2813.

[81] Y. Zhou, J. Li, H. Chen, Y. Wu, J. Wu, L. Chen, A spatiotemporal attention mechanism-based model for multi-step citywide passenger demand prediction, Information Sciences 513 (2020) 372–385.

[82] C. F. Daganzo, Y. Ouyang, A general model of demand-responsive transportation services: From taxi to ridesharing to dial-a-ride, Transportation Research Part B: Methodological 126 (2019) 213–224.

[83] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, Y. Liu, Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 3656–3663.

[84] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang, Traffic flow prediction with big data: a deep learning approach, IEEE Transactions on Intelligent Transportation Systems 16 (2) (2014) 865–873.

[85] J. Baek and, K. Sohn, Deep-learning architectures to forecast bus ridership at the stop and stop-to-stop levels for dense and crowded bus networks, Applied Artificial Intelligence 30 (9) (2016) 861–885.

[86] Y. Duan, Y. Lv, Y.-L. Liu, F.-Y. Wang, An efficient realization of deep learning for traffic data imputation, Transportation research part C: emerging technologies 72 (2016) 168–181.

[87] A. Koesdwiady, R. Soua, F. Karray, Improving traffic flow prediction with weather information in connected cars: A deep learning approach, IEEE Transactions on Vehicular Technology 65 (12) (2016) 9508–9517.

[88] H.-F. Yang, T. S. Dillon, Y.-P. P. Chen, Optimized structure of the traffic flow forecasting model with a deep learning approach, IEEE transactions on neural networks and learning systems 28 (10) (2016) 2371–2381.

[89] Y. Jia, J. Wu, Y. Du, Traffic speed prediction using deep learning method, in: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2016, pp. 1217–1222.

[90] Y. Jia, J. Wu, M. Xu, Traffic flow prediction with rainfall impact using a deep learning method, Journal of advanced transportation 2017 (2017).

[91] T. Pamuła, Impact of data loss for prediction of traffic flow on an urban road using neural networks, IEEE Transactions on Intelligent Transportation Systems 20 (3) (2018) 1000–1009.

[92] Y. Zhang, G. Huang, Traffic flow prediction model based on deep belief network and genetic algorithm, IET Intelligent Transport Systems 12 (6) (2018) 533–541.

[93] Y. Tian, L. Pan, Predicting short-term traffic flow by long short-term memory recurrent neural network, in: 2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity), IEEE, 2015, pp. 153–158.

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

[95] R. Fu, Z. Zhang, L. Li, Using lstm and gru neural network methods for traffic flow prediction, in: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), IEEE, 2016, pp. 324–328.

[96] M. Elhenawy, H. Rakha, Stretch-wide traffic state prediction using discriminatively pre-trained deep neural networks, in: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2016, pp. 1065–1070.

[97] D. Wang, W. Cao, J. Li, J. Ye, Deepsd: Supply-demand prediction for online car-hailing services using deep neural networks, in: 2017 IEEE 33rd international conference on data engineering (ICDE), IEEE, 2017, pp. 243–254.

[98] A. Sudo, T.-H. Teng, H. C. Lau, Y. Sekimoto, Predicting indoor crowd density using column-structured deep neural network, in: Proceedings of the 1st ACM SIGSPATIAL Workshop on Prediction of Human Mobility, 2017, pp. 1–7.

[99] Z. Yuan, X. Zhou, T. Yang, J. Tamerius, R. Mantilla, Predicting traffic accidents through heterogeneous urban data: A case study, in: Proceedings of the 6th International Workshop on Urban Computing (UrbComp 2017), Halifax, NS, Canada, Vol. 14, 2017.

[100] X. Dai, R. Fu, Y. Lin, L. Li, F.-Y. Wang, Deeptrend: A deep hierarchical neural network for traffic flow prediction, arXiv preprint arXiv:1707.03213 (2017).

[101] D. Kang, Y. Lv, Y.-y. Chen, Short-term traffic flow prediction with lstm recurrent neural network, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2017, pp. 1–6.

[102] Z. Zhao, W. Chen, X. Wu, P. C. Chen, J. Liu, Lstm network: a deep learning approach for short-term traffic forecast, IET Intelligent Transport Systems 11 (2) (2017) 68–75.

[103] Y. Jia, J. Wu, M. Ben-Akiva, R. Seshadri, Y. Du, Rainfall-integrated traffic speed prediction using deep learning method, IET Intelligent Transport Systems 11 (9) (2017) 531–536.

[104] N. G. Polson, V. O. Sokolov, Deep learning for short-term traffic flow prediction, Transportation Research Part C: Emerging Technologies 79 (2017) 1–17.

[105] F. Sun, A. Dubey, J. White, Dxnat - deep neural networks for explaining non-recurring traffic congestion, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE, 2017, pp. 2141–2150.

[106] C. Xu, J. Ji, P. Liu, The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets, Transportation research part C: emerging technologies 95 (2018) 47–60.

[107] Z. Abbas, A. Al-Shishtawy, S. Girdzijauskas, V. Vlassov, Short-term traffic prediction using long short-term memory neural networks, in: 2018 IEEE International Congress on Big Data (BigData Congress), IEEE, 2018, pp. 57–65.

[108] Z. Xie, Q. Liu, Lstm networks for vessel traffic flow prediction in inland waterway, in: 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), IEEE, 2018, pp. 418–425.

[109] D. Zhang, M. R. Kabuka, Combining weather condition data to predict traffic flow: a gru-based deep learning approach, IET Intelligent Transport Systems 12 (7) (2018) 578–585.

[110] B. Zhao, X. Zhang, A parallel-res gru architecture and its application to road network traffic flow forecasting, in: Proceedings of 2018 International Conference on Big Data Technologies, 2018, pp. 79–83.

[111] Y. Adu-Gyamfi, M. Zhao, Traffic speed prediction for urban arterial roads using deep neural networks, Journal of Computing in Civil Engineering (2019).

[112] Z. Cui, R. Ke, Z. Pu, Y. Wang, Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction, arXiv preprint arXiv:1801.02143 (2018).

[113] A. Fandango, R. P. Wiegand, Towards investigation of iterative strategy for data mining of short-term traffic flow with recurrent neural networks, in: Proceedings of the 2nd International Conference on Information System and Data Mining, 2018, pp. 65–69.

[114] B. Liao, J. Zhang, M. Cai, S. Tang, Y. Gao, C. Wu, S. Yang, W. Zhu, Y. Guo, F. Wu, Dest-resnet: A deep spatiotemporal residual network for hotspot traffic speed prediction, in: Proceedings of the 26th ACM international conference on Multimedia, 2018, pp. 1883–1891.

[115] W. Xiangxue, X. Lunhui, C. Kaixun, Data-driven short-term forecasting for urban road network traffic based on data processing and lstm-rnn, Arabian Journal for Science and Engineering 44 (4) (2019) 3043–3060.

[116] S. Sun, J. Chen, J. Sun, Traffic congestion prediction based on gps trajectory data, International Journal of Distributed Sensor Networks 15 (5) (2019) 1550147719847440.

[117] Y. Wu, H. Tan, Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework, arXiv preprint arXiv:1612.01022 (2016).

[118] S. Du, T. Li, X. Gong, Y. Yang, S. J. Horng, Traffic flow forecasting based on hybrid deep learning framework, in: 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), IEEE, 2017, pp. 1–6.

[119] Y. Liu, H. Zheng, X. Feng, Z. Chen, Short-term traffic flow prediction with conv-lstm, in: 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP), IEEE, 2017, pp. 1–6.

[120] H. Yu, Z. Wu, S. Wang, Y. Wang, X. Ma, Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks, Sensors 17 (7) (2017) 1501.

[121] G. Yang, Y. Wang, H. Yu, Y. Ren, J. Xie, Short-term traffic state prediction based on the spatiotemporal features of critical road sections, Sensors 18 (7) (2018) 2287.

[122] Z. Duan, Y. Yang, K. Zhang, Y. Ni, S. Bajgain, Improved deep hybrid networks for urban traffic flow prediction using trajectory data, IEEE Access 6 (2018) 31820–31827.

[123] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, arXiv preprint arXiv:1709.04875 (2017).

[124] X. Cheng, R. Zhang, J. Zhou, W. Xu, Deeptransport: Learning spatial-temporal dependency for traffic condition forecasting, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.

[125] X. Ning, L. Yao, X. Wang, B. Benatallah, F. Salim, P. D. Haghighi, Predicting citywide passenger demand via reinforcement learning from spatio-temporal dynamics, in: Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2018, pp. 19–28.

[126] F. Rodrigues, I. Markou, F. C. Pereira, Combining time-series and textual data for taxi demand prediction in event areas: A deep learning approach, Information Fusion 49 (2019) 120–129.

[127] Y. Li, Y. Zheng, Q. Yang, Dynamic bike reposition: A spatio-temporal reinforcement learning approach, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1724–1733.

[128] L. Wang, X. Geng, X. Ma, F. Liu, Q. Yang, Cross-city transfer learning for deep spatio-temporal prediction, arXiv preprint arXiv:1802.00386 (2018).

[129] H. Yao, X. Tang, H. Wei, G. Zheng, Y. Yu, Z. Li, Modeling spatial-temporal dynamics for traffic prediction, arXiv preprint arXiv:1803.01254 (2018).

[130] S. Du, T. Li, X. Gong, Z. Yu, Y. Huang, S.-J. Horng, A hybrid method for traffic flow forecasting using multimodal deep learning, arXiv preprint arXiv:1803.02099 (2018).

[131] Y. Liang, Z. Cui, Y. Tian, H. Chen, Y. Wang, A deep generative adversarial architecture for network-wide spatial-temporal traffic-state estimation, Transportation Research Record 2672 (45) (2018) 87–105.

[132] T. Wu, F. Chen, Y. Wan, Graph attention lstm network: A new model for traffic flow forecasting, in: 2018 5th International Conference on Information Science and Control Engineering (ICISCE), IEEE, 2018, pp. 241–245.

[133] Y. Wu, H. Tan, L. Qin, B. Ran, Z. Jiang, A hybrid deep learning based traffic flow prediction method and its understanding, Transportation Research Part C: Emerging Technologies 90 (2018) 166–180.

[134] Q. Liu, B. Wang, Y. Zhu, Short-term traffic speed forecasting based on attention convolutional neural network for arterials, Computer-Aided Civil and Infrastructure Engineering 33 (11) (2018) 999–1016.

[135] X. Ma, Y. Li, Z. Cui, Y. Wang, Forecasting transportation network speed using deep capsule networks with nested lstm models, arXiv preprint arXiv:1811.04745 (2018).

[136] Z. He, C.-Y. Chow, J.-D. Zhang, Stann: A spatio–temporal attentive neural network for traffic prediction, IEEE Access 7 (2018) 4795–4806.

[137] L. Mourad, H. Qi, Y. Shen, B. Yin, Astir: Spatio-temporal data mining for crowd flow prediction, IEEE Access 7 (2019) 175159–175165.

[138] Y. Zhou, H. Chen, J. Li, Y. Wu, J. Wu, L. Chen, St-attn: Spatial-temporal attention mechanism for multi-step citywide crowd flow prediction, in: 2019 International Conference on Data Mining Workshops (ICDMW), IEEE, 2019, pp. 609–614.

[139] Y. Zhang, H. Wang, D. Zhang, D. Wang, Deeprisk: A deep transfer learning approach to migratable traffic risk estimation in intelligent transportation using social sensing, in: 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), IEEE, 2019, pp. 123–130.

[140] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 922–929.

[141] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, J. Zhang, Urban traffic prediction from spatio-temporal data using deep meta learning, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 1720–1730.

[142] C. Zheng, X. Fan, C. Wang, J. Qi, Gman: A graph multi-attention network for traffic prediction, arXiv preprint arXiv:1911.08415 (2019).

[143] C. Zhang, J. James, Y. Liu, Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting, IEEE Access 7 (2019) 166246–166256.

[144] Y. Zhang, S. Wang, B. Chen, J. Cao, Z. Huang, Trafficgan: Network-scale deep traffic prediction with generative adversarial nets, IEEE Transactions on Intelligent Transportation Systems (2019).

[145] X. Yi, Z. Duan, T. Li, T. Li, J. Zhang, Y. Zheng, Citytraffic: Modeling citywide traffic via neural memorization and generalization approach, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 2665–2671.

[146] L. Cai, K. Janowicz, G. Mai, B. Yan, R. Zhu, Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting, Transactions in GIS.

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

[148] https://ai.baidu.com/broad/download?dataset=traffic.

[149] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

[150] https://github.com/TolicWang/DeepST/tree/master/data/TaxiBJ.

[151] https://www.citibikenyc.com/system-data.

[152] http://web.mta.info/developers/MTA-Bus-Time-historical-data.html.

[153] https://data.gov.uk/dataset/9562c512-4a0b-45ee-b6ad-afc0f99b841f/\highways-england-network-journey-time-a

[154] HTTP://wsdot.uwdrive.net/STARLab.

[155] https://archive.ics.uci.edu/ml/datasets/Taxi+Service+Trajectory+-+\Prediction+Challenge,+ECML+PKDD+2015.

[156] https://outreach.didichuxing.com/appEn-vue/personal?id=2.

[157] https://www.capitalbikeshare.com/system-data.

[158] https://www.divvybikes.com/system-data.

[159] https://resource.data.one.gov.hk/td/speedmap.xml.

[160] https://www.kaggle.com/fivethirtyeight/uber-pickups-in-new-york-city.

[161] https://topis.seoul.go.kr/.

[162] https://data.cityofchicago.org/Transportation/Chicago-Traffic-Tracker-Historical-Congestion-Esti/77hq-huss/data.

[163] D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, IEEE transactions on evolutionary computation 1 (1) (1997) 67–82.

[164] M. Noruzoliaee, B. Zou, Y. Liu, Roads in transition: Integrated modeling of a manufacturer-traveler-infrastructure system in a mixed autonomous/human driving environment, Transportation Research Part C: Emerging Technologies 90 (2018) 307–333.