Transportation Research Part C 133 (2021) 103432

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Review

Applications of deep learning in congestion detection, prediction and alleviation: A survey

Nishant Kumar [a,*], Martin Raubal [b,a]

[a] ETH Zurich, Future Resilient Systems, Singapore-ETH Centre, 1 CREATE Way, #06-01 CREATE Tower, Singapore 138602, Singapore
[b] Institute of Cartography and Geoinformation, ETH Zurich, Stefano-Franscini-Platz 5, 8093 Zurich, Switzerland

ARTICLE INFO

Keywords:
Deep learning
Transportation
Congestion
Recurring
Non-recurring
Accidents

ABSTRACT

Detecting, predicting, and alleviating traffic congestion are targeted at improving the level of service of the transportation network. With increasing access to larger datasets of higher resolution, the relevance of deep learning for such tasks is increasing. Several comprehensive survey papers in recent years have summarised the deep learning applications in the transportation domain. However, the system dynamics of the transportation network vary greatly between the non-congested state and the congested state — thereby necessitating the need for a clear understanding of the challenges specific to congestion prediction. In this survey, we present the current state of deep learning applications in the tasks related to detection, prediction, and alleviation of congestion. Recurring and non-recurring congestion are discussed separately. Our survey leads us to uncover inherent challenges and gaps in the current state of research. Finally, we present some suggestions for future research directions as answers to the identified challenges.

1. Introduction

Traffic congestion decreases the level of serviceability (LOS) of road networks. A decrease in LOS results in direct and indirect costs to society. Extensive studies have been carried out to estimate the impacts of congestion on the economy and society as a whole (Weisbrod et al., 2001; Litman, 2016). The first-hand impact of traffic congestion is the lost working hours. Schrank et al. (2012) estimated that in a single year, the USA alone lost a total of 8.8 billion working hours due to congestion. The detrimental impacts of congestion skyrocket when the value of time, as a commodity, increases drastically during emergencies. Being stuck in traffic impacts the behaviour of individuals. Hennessy and Wiesenthal (1999) report that high congestion levels can result in aggressive behaviour by drivers. This aggression can manifest itself into aggressive driving, thereby increasing the chances of accidents (Li et al., 2020a). High levels of congestion also result in higher greenhouse gas emissions (Barth and Boriboonsomsin, 2009).

In terms of tractability, congestion prediction is a more difficult problem than traffic prediction during uncongested conditions (Yu et al., 2017). An early warning system enables traffic controllers to put alleviation measures in place. The infrastructure required for traffic data collection has improved over the decades. This improvement, in conjunction with the increased availability of computational resources, has enabled transportation researchers to leverage the predictive capabilities of deep neural networks for this domain. In this survey, we discuss the applications of deep learning in the detection, prediction, and alleviation of congestion. We investigate various aspects of the two types of congestion — recurring and non-recurring. Towards the end of this survey, we identify some gaps in the current state of the research in this field and present future research directions.

* Corresponding author.
E-mail addresses: niskumar@ethz.ch (N. Kumar), mraubal@ethz.ch (M. Raubal).

https://doi.org/10.1016/j.trc.2021.103432
Received 4 February 2021; Received in revised form 4 September 2021; Accepted 8 October 2021
Available online 1 November 2021
0968-090X/© 2021 Singapore-ETH Centre. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

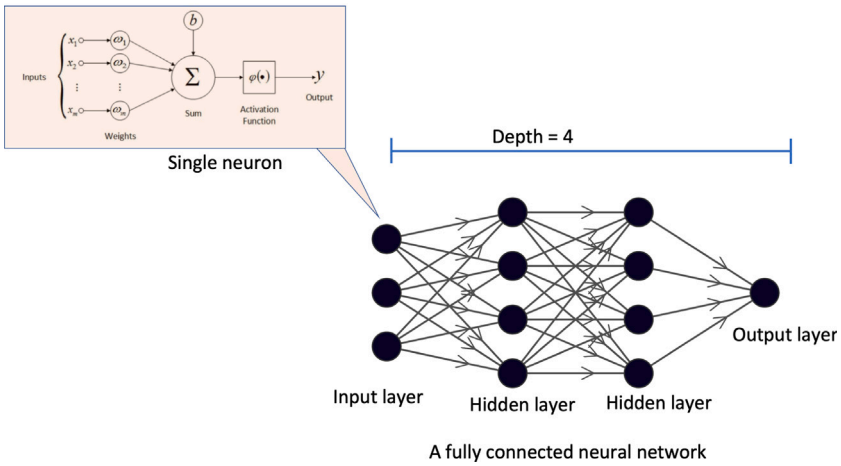

Fig. 1. A fully connected layered neural network with two hidden layers. The term 'layered' is often omitted in the literature. In this survey, we refer to these as feed forward neural networks. An enlarged diagram of a single artificial neuron is presented separately to show its three components — inputs, weighted sum, and nonlinear activation. The input data point has n features (also called dimensions).
Source: Single neuron image obtained from Chrislb — Own work, https://commons.wikimedia.org/w/index.php?curid=224550[wikimedia public domain CC BY-SA 3.0)]. The layered network created using Nn-svg (LeNail, 2019) .

2. Preliminaries

The target audience of this survey paper are researchers from two backgrounds- transportation and deep learning. In the following two subsections, we cover the preliminaries and introduce the terminology which is used throughout the survey.

2.1. Relevant concepts and terms in deep learning

An artificial neuron is a function as shown in Eq. (1)

$$y_i = \varphi\left(\sum_{j=0}^{m} w_{ij}x_{ij}\right) \tag{1}$$

where x_{ij} is the jth feature (dimension) of the ith m-dimensional data point in the dataset; w_j (called weights) is the coefficient which is tuned during the training process of the neural network; φ is a nonlinear activation function; y_i is the output of the function on input x_i. Commonly used activation functions are: sigmoid ($\varphi(x) = \frac{1}{1+e^{-x}}$), tanh ($\varphi(x) = \frac{e^{2x}-1}{e^{2x}+1}$) and relu ($\varphi(x) = max(x, 0)$).

2.1.1. Fully connected layered neural networks

The network shown in Fig. 1 is a fully connected layered neural network. In such neural networks, the outputs from all neurons from a previous layer are fed as inputs to all neurons in the next layer. The literature on neural networks often omits the term 'layered' and uses the terms fully connected neural network (FCNN) in its place. The usage can be a source of confusion because a fully connected network might imply the presence of connections between all neurons in the neural network, not just between layers placed next to each other. In order to ensure proper terminology, in this survey we stick to the term feed forward neural network (FFNN) to imply fully connected layered neural networks (Goodfellow et al., 2016, Chapter 6). Additionally, we observed that several papers referred to in this survey used the term artificial neural network (ANN) to denote FFNN. In this survey, however, we stick to FFNN and avoid using ANN in its place.

When placed parallel to each other, several such neurons form a *layer* of the neural network. When several layers are stacked one after the other, a feed forward neural network (FFNN) is formed. In this context, stacking refers to passing the output of one function or unit to another. The formation of a neural network using neurons is shown in Fig. 1. As we increase the stacking, the *depth* of the neural network increases. The literature on neural networks does not specify a pre-defined threshold for the depth, in order to demarcate *deep* and *shallow* networks. Any neural network having more than one hidden layer can be referred to as a deep neural network (Schmidhuber, 2015). A deep neural network can learn more abstract representations of the data compared to a shallow network. The link between depth and abstractions is easily observed when working with image data as shown in Fig. 2. We shall frequently use the terms depth and layers in this survey.

For supervised learning tasks using deep learning, such as the prediction of congestion, the goal is to train a deep learning model so that it learns a mapping from input data to the output data. Let us consider a traffic prediction task where the goal is to predict the traffic flow at n locations 1 min into the future. The input data x is a vector of length n and varies at every minute t. Training the deep learning model implies assuming the existence of an underlying function f such that $x(t+1) = f(x(t))$ and then, attempt to approximate f by adapting the weights of the model. In order to approximate the function $f(x)$, a loss function L is

minimised. The most commonly used loss function is mean squared error. The minimisation is carried out using the backpropagation algorithm (LeCun et al., 1988). Backpropagation refers to the propagation of errors from the loss function to the previous layers using the chain rule of differentiation. In order to guarantee that the function approximated by the deep learning model is not arbitrary, the loss is computed on new unseen data (test data) after the training process. If the values of loss function on training data and test data are similar, the model is said to generalise well. Several techniques for better generalisation of deep learning models have been explored. The most commonly used technique is dropout (Srivastava et al., 2014). In this survey, we pay special attention to the generalisation techniques which have been developed by leveraging upon the domain knowledge from transportation.

2.1.2. CNNs and RNNs

Traffic data vary over space and time. Two families of neural network architectures are particularly suited to capture such inter-dependencies: CNN and RNN.

Convolutional Neural Networks: CNN stands for Convolutional Neural Network. Historically, CNNs have been popularly used in image-classification problems due to their ability to capture the correlation between nearby pixels of an image. A deep-CNN is able to capture the correlations between pixels placed far apart in the image. In a typical CNN architecture, the first few layers are convolutional blocks, interspersed with pooling layers. Fully connected layers are present just before the output layer. Pooling is a downsampling technique used to report summary statistics from a neighbourhood (Goodfellow et al., 2016, chap. 9). The most commonly used pooling method with CNNs is max-pooling, wherein the maximum value of the activation is selected from a region (Albawi et al., 2017). Pooling helps reduce the complexity of the deep learning model and also learn representations that are invariant to small local translations of the input data. The effectiveness of deep CNNs in capturing the spatial dependencies in the image is illustrated in Fig. 2. When applied to traffic data such as a 2-D image of grid-wise congestion level, the CNN models can capture the spatial dependencies.

Recurrent Neural Networks: Based on the previous example, if we extend our assumed underlying function f to be dependent on the previous 10 time steps instead of just one, $(x(t+1) = f(x(t), x(t-1), x(t-2), x(t-3), ..x(t-10)))$, training an FFNN can be achieved by concatenating sequences of 10 input vectors to create input vectors $(x'(t))$ of length $(n * 10)$. The other option is to do backpropagation through time. This is accomplished using Recurrent Neural Networks (RNNs) as shown in Fig. 4. A RNN has a feedback loop in the connection, as opposed to the feed forward neural networks. The feedback loop can be unravelled to uncover the back propagation in time. When time series data are passed in sequence as an input to a RNN, the RNN maintains an internal state from one time-step to the next. At time $t+1$, the hidden state is influenced by the input at time t and the previous hidden state. This helps the RNNs to unravel the temporal dependencies in the data. Long-short-term-memory (LSTM) introduced in Hochreiter and Schmidhuber (1997), is an improvement over the traditional RNN. LSTM networks can detect dependencies between data points which are far apart in time (Greff et al., 2016). At time t, an LSTM cell is characterised by the state of four logic gates — input (i_t), output (o_t), cell state (c_t) and forget (f_t) gates. Using our example of the vector $x(t)$, the hidden state (h_t) of an LSTM unit can be formalised using Eqs. (2)–(6) (Tian et al., 2018; Wang et al., 2019).

$$i_t = \sigma\left(W_{\mathrm{xi}}x_t + W_{\mathrm{hi}}h_{t-1}\right) \tag{2}$$

$$f_t = \sigma\left(W_{\mathrm{xf}}x_t + W_{\mathrm{hf}}h_{t-1}\right) \tag{3}$$

$$o_t = \tanh\left(W_{\mathrm{xo}}x_t + W_{\mathrm{ho}}h_{t-1}\right) \tag{4}$$

$$c_t = c_{t-1} \odot f_t + i_t \odot \tanh\left(W_{\mathrm{xc}}x_t + W_{\mathrm{hc}}h_{t-1}\right) \tag{5}$$

$$h_t = \tanh\left(c_t\right) \odot o_t \tag{6}$$

where the W_{ab} refers to the weight matrix between gates a and b, $\odot$ refers to element-wise vector product, h_t refers to the hidden state at time t, i_t refers to the input at time t, o_t refers to the output at time t and σ refers to the sigmoid activation function.

Another recent improvement over LSTM are the Gated Recurrent Units, proposed in Chung et al. (2014). Compared to LSTM, GRU has a less complex structure and can be trained faster than LSTM. At time t, a GRU cell is characterised by the state of two logic gates — update gate (z_t) and reset gate (r_t). For a detailed empirical comparison description of the differences between RNNs and LSTMs, the interested reader is referred to Jozefowicz et al. (2015) for an empirical evaluation of GRUs and LSTMs. The hidden state of a GRU can be formalised using Eqs. (7)–(9) (Wang et al., 2019):

$$r_t = \sigma\left(W_{\mathrm{xr}}x_t + W_{\mathrm{hr}}h_{t-1}\right) \tag{7}$$

$$z_t = \sigma\left(W_{\mathrm{xz}}x_t + W_{\mathrm{hz}}h_{t-1}\right) \tag{8}$$

$$h_t = z_t \odot h_{t-1} + \left(1 - z_t\right) \odot \left(\tanh\left(W_{\mathrm{xh}}x_t + W_{\mathrm{hh}}\left(r_t \odot h_{t-1}\right)\right)\right) \tag{9}$$

where the W_{ab} refers to the weight matrix between gates a and b, $\odot$ refers to element-wise vector product, h_t refers to the hidden state at time t and σ refers to the sigmoid activation function.

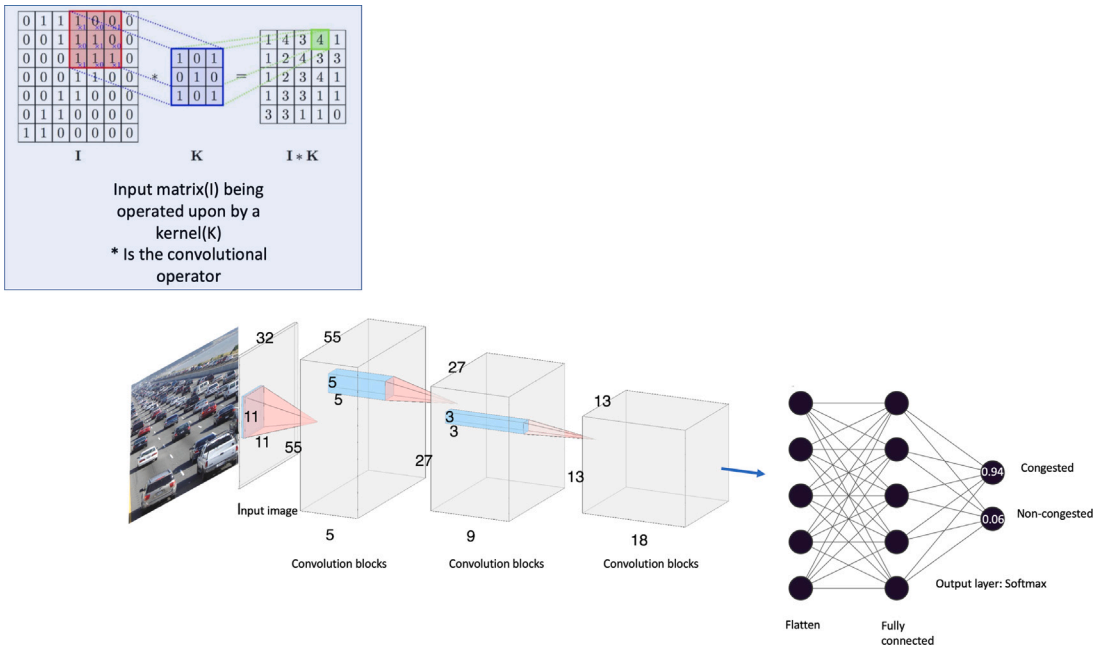

Fig. 2. (Top: A single convolution operation; Bottom: A deep convolutional neural network (CNN). As data propagate through various convolutional layers, the features learnt become more high level.
Source: Single convolution image adapted from Mohamed (2017) with permission, traffic jam image adapted from https://upload.wikimedia.org/wikipedia/commons/3/3e/I-80_Eastshore_Fwy.jpgwikimedia public domain CC BY-SA 3.0).

2.1.3. Reinforcement learning

Apart from CNNs and RNNs, a commonly used deep learning framework for traffic prediction tasks is deep reinforcement learning. Reinforcement learning (RL) is a learning paradigm that, when combined with deep learning, serves as a powerful tool for specific traffic-related prediction tasks where *control* is involved. Deep-reinforcement learning models have been demonstrated to perform very well for specific tasks — the most notable example is the model that was able to learn the game of *Go* starting from scratch (basic game rules) to a level that surpassed the rating of world champions (Silver et al., 2017). The high computational load of such models, however, restricts their wide use. Szepesvári (2010, Page-1, abstract) define reinforcement learning as "a learning paradigm concerned with learning *to control* a system so as to maximise a numerical performance measure that expresses a long-term objective". In this context, *how to control* is also referred to as the *policy* being learnt. When a deep learning model is trained to learn the best policy, it is referred to as a deep-reinforcement learning model. A representation of the reinforcement learning framework is presented in Fig. 5. While reviewing the literature, we found that Q-learning appears to be the popular reinforcement learning framework for traffic prediction tasks. Q-learning, introduced in Watkins (1989), is a model-free reinforcement learning approach where the *environment* as shown in Fig. 5, does not need to be modelled explicitly. A detailed discussion on DQN is presented in Mnih et al. (2015).

2.1.4. Commonly used metrics

The commonly used metrics to quantify performance for regression tasks are Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE), given by:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}^i - y^i\right| \tag{10}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}^i - y^i\right)^2} \tag{11}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}^i - y^i}{y^i}\right| * 100\% \tag{12}$$

where $\hat{y}^i$ is the prediction for the ith data point, where the ground truth value was y^i. As is evident from the equations, RMSE and MAE are unit dependent while MAPE is a dimensionless quantity. In this survey, while reporting the performance of various regression tasks, we have tried to report the MAPE whenever available.

		True Class	
		Congested	Non-congested
Predicted Class	Congested	True Positive (TP) Congested predicted as Congested	False Positive (FP) Non-congested predicted as congested
	Non-congested	False Negative (FN) Congested predicted as Non-congested	True Negative (TN) Non-congested predicted as Non-congested

Fig. 3. Confusion matrix for binary classification task.

The commonly used metrics for classification tasks can be summarised using a confusion matrix as shown in Fig. 3. In the light of the confusion matrix, several metrics are defined. The most commonly used metrics are: true positive rate (TPR), true negative rate (TNR) and accuracy, given by:

$$TPR = \frac{TP}{TP + FN} \tag{13}$$

$$TNR = \frac{TN}{TN + FP} \tag{14}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

The choice of the metric used to evaluate the performance is often determined by the task at hand. For example, let us consider an example where a deep learning model is being used to classify whether the traffic state is 'congested' or 'not congested. Let us consider 1440 data points collected every minute over the course of 24 h and the congestion lasted for an hour and 60 data points have the ground truth label as 'congestion'. This implies that even if the model predicts all data points as 'not congested', the prediction accuracy is $(\frac{0+(1440-60)}{1440} = 95.8\%$. Thus, total accuracy is a misleading term for congestion prediction tasks. So, for congestion prediction tasks, the usual practice is to report balanced accuracy (BAC). BAC is defined as the mean of the TPR for each class $(0.5 * (TPR + TNR))$. We observed that some papers presented here refer to BAC as *average accuracy*. Another metric used by one of the papers discussed here is quadratic weighted kappa (QWK Ben-David, 2008). The QWK metric increases the penalty for classification by chance. The QWK values lie between 0 and 1, with 1 being achieved when the prediction matches the ground truth and 0 when the output is random noise. Mathematically, for an N class classification task, QWK is given by $\kappa = 1 - \frac{\Sigma_{i,j} \mathbf{w}_{i,j} \mathbf{O}_{i,j}}{\Sigma_{i,j} \mathbf{E}_{i,j} \mathbf{P}_{i,j}}$, where $\mathbf{O_{NxN}}$ is the confusion matrix according to the model prediction and the matrix $\mathbf{E}_{NxN}$ is the expected confusion matrix if prediction was by chance; $\mathbf{w}_{i,j} = \frac{(i-j)^2}{(N-1)^2}$ A complete derivation is presented in Haberman (2019).

The choice of the metric is an important design decision in machine learning tasks and influences the performance of the modelling task. Some common factors which are taken into account while choosing the metric are class imbalance, presence of outliers and invariance properties. The interested reader is referred to Sokolova and Lapalme (2009) for a systematic coverage of various metrics for classification tasks.

2.2. Relevant concepts and terms in transportation

In this section, we define some terms from transportation that are fundamental to the understanding of the discussion presented in this survey. A comprehensive review of these definitions and deep insights on their role in traffic prediction is presented in Hall (1996), Immers and Logghe (2002), and Gerlough and Huber (1976, Chapter 2). Here, we selectively reproduce some basic ideas which are necessary for the discussion presented in this survey. The most common variables used to measure the state of traffic are *density*, *speed* and *flow*.

Traffic density: Traffic density is defined as the number of vehicles per unit length of the road segment. Traditionally, traffic density for the entire road segment has been difficult to measure because of the limited number of sensors to estimate the presence of vehicles. However, this trend is changing with an increasing number of traffic cameras and the advances in computer vision. A related quantity is 'occupancy', which is often used as a proxy for measuring density. Occupancy is defined as the percentage of the time during which a point in the road network is occupied by vehicles. Occupancy can be directly measured using sensors, most commonly using Vehicle loop detectors (VLDs). If the length of each vehicle is the same (homogeneous stream of traffic), occupancy is directly proportional to the traffic flow. In practice, the density is most commonly estimated using the fundamental

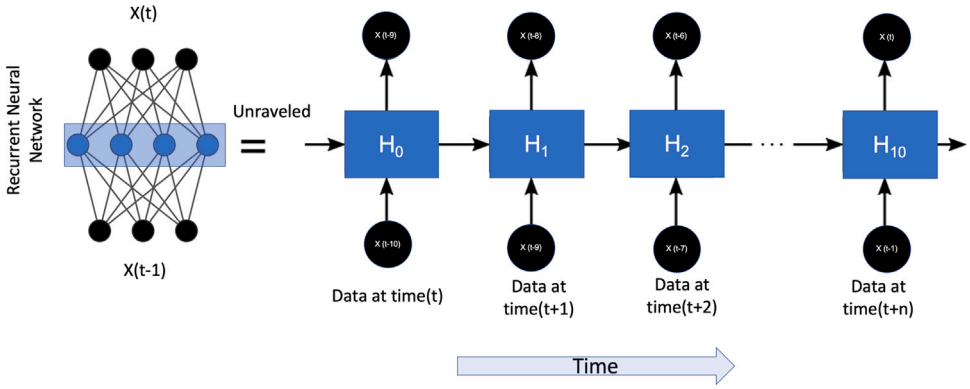

Fig. 4. Unravelling of a recurrent neural network in time.

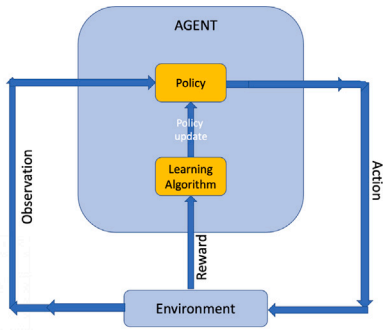

Fig. 5. The general framework of reinforcement learning.

relation between density and speed ($q = k * u$), where q is the flow, k is the density and u is the speed. In the presence of a heterogeneous traffic stream, the relationship between occupancy and density is complex (Ramezani et al., 2015).

Traffic speed: The spot speed (or instantaneous speed) of a vehicle is the speed that is recorded at a given moment in time and at a specified location. This is the speed that is measured on the vehicle speedometer. In transportation engineering, however, we are interested in determining mean speed which can be used as a defining parameter for the traffic stream. In order to compute mean speed, the aggregation can be done in time or space. Space–mean speed, for a given interval of space, is defined as the ratio between the total distance travelled by all vehicles and the total time taken. Time–mean speed, for a given interval of time, is defined as the arithmetic mean of the individual speed of all vehicles. Mathematically, the space–mean speed reduces to the harmonic mean of individual vehicle speeds (v_i). Assuming N vehicles,

$$space\ mean\ speed = \frac{total\ distance\ travelled}{total\ time\ taken} = \frac{N * D}{\sum_i^N t_i} = \frac{N * D}{\sum_i^N \frac{D}{u_i}} = \frac{1}{N}\frac{1}{\sum_i^N \frac{1}{u_i}} \tag{16}$$

$$time\ mean\ speed = \frac{1}{N}\sum_i N u_i \tag{17}$$

The space–mean speed satisfies the fundamental relation between flow, speed and density ($q = k * u$), whereas the time–mean speed does *not* follow the fundamental equation. When using deep learning for predicting traffic speed, the pre-processing steps on the raw data determine which one of them is being predicted. The use of space–mean speed is more common for congestion prediction tasks commonly referred to as the 'segment speed. On the other hand, if the data source provides aggregated speed data, the general practice is to predict the same variable (Gartner et al., 2002; Daganzo and Daganzo, 1997).

Traffic flow: Traffic flow is defined as the number of vehicles passing a reference point per unit time. The reference points are usually chosen in the middle or at the end of a segment.

Travel time: Travel time is the time taken by a vehicle to go from point A to point B. Traditionally, travel time has been difficult to measure using aggregated data from point sensors (such as VLDs). With the advent of distributed sensors, such as GPS, these are being increasingly used to estimate the travel time. In the transportation literature, this is referred to as the floating car data (FCD). The challenge however lies in the variation in the percentage of vehicles that share the data at any given point. The major

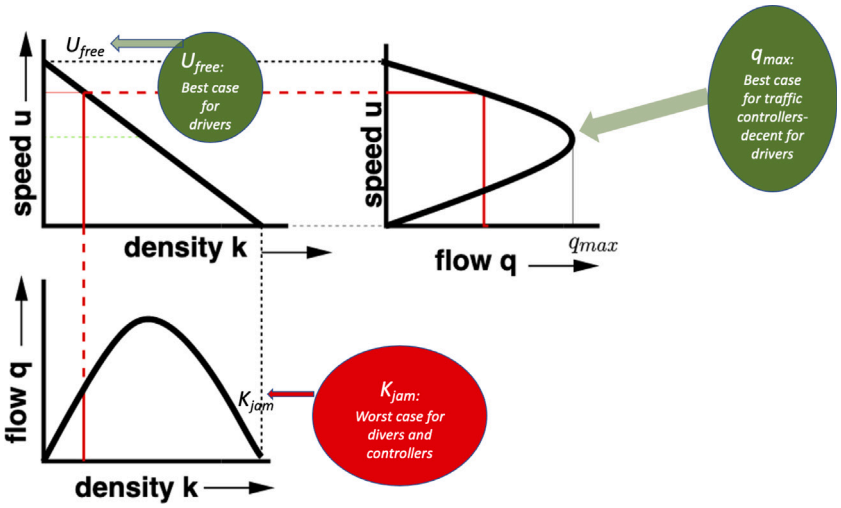

Fig. 6. Representative curves showing the relationship between flow, speed and density. A linear relation between speed and density is assumed here. The two green circles point to maximum speed (U_{max}) and maximum flow (q_{max}) which are the best serviceability conditions w.r.t drivers and traffic controllers respectively. The red circle points to the maximum density or traffic jam (K_{jam}), which implies a lack of serviceability from both perspectives..
Source: Base image adapted from NPTEL lecture notes creative commons license (CC-BY-NC-SA).

benefit of using FCD is that when the traffic flow is high, more data are collected. This stands in contrast to point sensors where the optimal choice of sensor locations is a major challenge (De Fabritiis et al., 2008).

Two extreme values are important to study the relationship between the aforementioned traffic state variables. Familiarity with these extreme values is necessary to understand the remaining part of this survey and other research dealing with disruptions in road networks. These two values are:

- jam density (k_j): The highest possible value of traffic density; this corresponds to traffic speed = 0 km/h.
- free-flow speed (u_f): The maximum speed at which the vehicles can travel on a given road segment. Under the assumption that drivers respect the speed limit, u_f is the same as the speed limit for the road segment under consideration.

The three variables described above (*speed*, *density* and *flow*) are correlated. However, a generalised equation depicting the relation between these variables has not been established. A simplified linear relationship between speed and density suffices for our discussion here. In Fig. 6, we show the relationship between these variables assuming a linear relationship between speed and density. The various critical points in the three curves in Fig. 6 are highlighted and colour-coded to show the level of serviceability for the two most important stakeholders in a transportation system — drivers and traffic controllers.

The choice of the target variable is also motivated by taking into account the consumers of the research output. If the research is targeted at optimising the usage of the transportation network as a system, the focus might be on maximising the throughput of the network; hence the researchers will focus on predicting the traffic flow accurately. On the other hand, if the research is aimed at improving the user travel time, the focus will be on predicting speed or travel time. For instance, when we use a trip planner to find the optimal route from a starting point to a destination, we often want to figure out the fastest route, we are not concerned with the traffic flow on the roads (Golledge, 1995).

Traffic hysteresis: The fundamental traffic diagram presented in Fig. 6, is overly simplified. When real data are plotted using density and speed (space–mean speed), the plot shows significant scatter around an underlying curve (Geroliminis and Sun, 2011). Researchers have proposed various theories to explain the scatter in the flow-density curve. One such theory is the theory of hysteresis characterised by a distinct loop in the flow-density curve. First observed by Treiterer (1975), traffic hysteresis arises due to the human factors in driving. The phenomenon can be easily observed at traffic intersections where a large number of vehicles are queuing. As soon as the signal turns green, the queue does not dissipate at the uniform rate throughout. The queue dissipates from front to back and there are significant time delays. The phenomenon of traffic hysteresis is attributed to the differential acceleration and deceleration rates of vehicles as shown in Fig. 7. As a result of the hysteresis, a traffic disrupting event continues to affect traffic even after the disruption event has ceased to occur.

Another theory used to explain the scatter is capacity drop, attributed to Cassidy and Bertini (1999). They propose that 'just before the onset of congestion the outflow out of a bottleneck is higher than in congestion' (van Wageningen-Kessels et al., 2015, p.451). The interested reader is referred to (van Wageningen-Kessels et al., 2015) for a detailed review of the traffic flow models. The key takeaways from these efforts are that the complexity of modelling traffic flow increases as the traffic state moves towards congestion.

Simulation-based approaches for traffic flow modelling: When using a microscopic agent-based traffic simulator, individuals and infrastructure elements are modelled as agents. Some examples of traffic simulators are: (1) open-sourced: MATSim (Axhausen

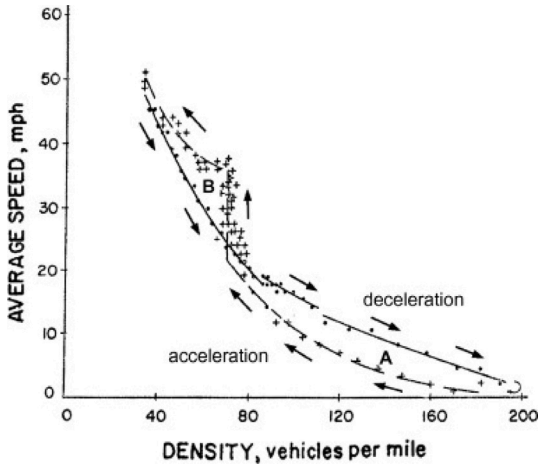

Fig. 7. Hysteresis loop. *Source:* Reproduced from Treiterer (1975).

et al., 2016), SUMO (Behrisch et al., 2011), SimMobility (Adnan et al., 2016), MATES (Yoshimura, 2006) and TRANSIMS (Smith et al., 1995; Nagel and Rickert, 2001) and (2) commercially available: AIMSUN (Casas et al., 2010), VISSIM and PARAMICS. In the case of traffic models derived using behavioural methods, a probabilistic model of the possible behaviours (actions and decisions) of each type of agent is programmed into the system by domain experts. The parameters are then calibrated using the available data. During the calibration process, the range of parameter values is constrained within meaningful ranges of values for each parameter. The initial values and the range of parameters, being set by the domain experts results in the parameters having some level of physical significance, thus making the models interpretable. The predicted traffic is the net result of the interaction between the agents in the calibrated model.

Short-term traffic prediction: Some model-based approaches to short-term traffic prediction are DynaMIT (Ben-Akiva et al., 1998) and DYNASMART (Mahmassani et al., 2004). In model-based approaches, the central algorithm is Dynamic Traffic Assignment (DTA) (Janson, 1991).

2.3. Definition and classification of congestion

'Congestion can be defined as the phenomenon that arises when the input volume exceeds the output capacity of a *facility*' (Stopher, 2004, Section 2.1). Depending on the number and size of *facilities*, congestion results in varying levels of loss in the serviceability of the road network. The literature on congestion prediction defines congestion either in terms of one of the traffic state variables (speed, density, flow) or in terms of derived variables such as the ratio of average speed to speed limit. A recent survey of the traffic variables used to define congestion is presented in Afrin and Yodo (2020). Once a variable is chosen, the values of the variable are quantified into a fixed number of levels in order to define a classification task. A binary quantisation can be achieved by using a single threshold on any one of the traffic variables. For example, a traffic density of more than a certain threshold can be referred to as 'jam' and vice-versa.

Based on the spatio-temporal frequency of their occurrences, congestion can be classified into two types — recurring and non-recurring. Recurring congestion, as the name suggests, is the congestion that manifests itself repeatedly in space and/or time. Specific areas of the city might experience traffic jams regularly at specific times during the day or on certain days of the week. Non-recurring congestion does not follow a spatio-temporal pattern. McGroarty (2010) present a summary of the causes behind both types of congestion. They report that recurring congestion is almost always caused by an infrastructural bottleneck. On the other hand, the non-recurring congestion can be caused by unforeseen events such as extreme weather conditions, natural and man-made disasters, accidents or planned events such as big concerts and roadworks. In the process of reviewing the literature, we observed that a significant number of papers have not clarified whether they attempted to predict recurring or non-recurring congestion. We have carefully evaluated their results and included them in their respective sections.

2.4. Synergies between model-based and deep learning based traffic prediction

The data-driven methods for short-term traffic prediction on the other hand do not take the user behaviour into consideration. Data-driven models aim to solve the prediction task by assuming traffic to be a measurable state of the system and attempt to predict its state into the future. Specifically, the workflow is to use all available data from sensors and output the predicted traffic state

variable. Particularly, when deep learning models are used for this task, the model internals (weights) have no physical significance. Due to the lack of interpretability, extensive validation is necessary to ensure that the deep learning model predictions are useful.

Traffic simulators are useful for investigating the effects of new policies. Their importance further increases in studies where real data are unavailable or cannot be collected. For example, a simulator can be used to study the effects of a city-wide failure of traffic lights. Real-world data cannot be obtained at such a scale; therefore the researchers rely on the fact that the behaviour-modelling of drivers and the modelled physical interactions between agents, together provide reliable inferences.

Synergies between model-based and data-driven approaches can benefit the research using both approaches. Congestion and accident databases suffer from severe class imbalance problems. Fukuda et al. (2020) used a traffic simulator to produce traffic data after simulated accidents. The generated data was then used for training a deep neural network. As we shall discuss in Section 6.1.1, when using a deep reinforcement learning framework to determine the optimal network-level control measures for congestion alleviation, a microscopic traffic simulator is incorporated in the framework. Borysov et al. (2019) used a deep generative model to generate agents for a simulation platform.

Deep learning models are increasingly being used to *learn* the physics behind the nonlinear dynamics of complex networks. Such models have typically been studied under the heading of Physics informed deep learning (PIDL)). PIDL was conceptualised with two motivations. First, PIDL enables us to use prior domain knowledge to regularise the function being approximated by the deep learning model, thereby reducing overfitting (Raissi et al., 2017a). Second, PIDL can be used to discover new partial differential equations from the data (Raissi et al., 2017b). When PIDL models are used in the traffic domain, microscopic traffic simulators often play an important role in the training process of such models. By design, the traffic simulators respect the traffic flow dynamics and hence can be used to regularise the traffic state predictions from a neural network. For example, SUMO was used for this purpose in Liu et al. (2020). The PIDL models hold a lot of potential because several drawbacks of deep learning models can be addressed. The PIDL models are more robust to missing data, noise, overfitting and might help in making deep learning models interpretable. However, the PIDL research in transportation is still at a nascent stage with very few papers and has not been covered in this survey. The interested reader is referred to Shi et al. (2021b), who used data from the Next Generation SIMulation (NGSIM) dataset from the US Department of Transportation and proposed a PIDL model with two components, one data-driven and the other, model-driven. The influence of each component in the training process can be controlled using a single parameter. In related work, Shi et al. (2021a) used a hybrid PIDL model on the same dataset to estimate the parameters of the second order partial differential equations governing traffic flow. Other recent efforts using PIDL are: Huang and Agarwal (2020), who present a detailed comparison between DL and PIDL models and report that PIDL models are faster to train (>50% faster) and perform better than other DL models when the sensor locations are fixed, Wang et al. (2020), who proposed a PIDL model called Turbulent-FlowNet for predicting turbulent traffic flow and Mo et al. (2020), who used a PIDL model to learn the dynamics of car-following models. To summarise, these efforts demonstrate promising avenues for further research into the synergies between model-based and data-driven techniques.

3. Previous surveys and organisation of this survey

The nonlinear activation functions in deep neural networks can capture the nonlinearities in the traffic data (Polson and Sokolov, 2017). As discussed in Section 2.1, the depth of the network enables us to model high-level features in the data. Traffic data are characterised by variations over space and time. Two specialised neural network architectures, CNN and RNN (also discussed in 2.1) are very helpful in capturing these variations. CNNs are useful while modelling spatial inter-dependencies whereas RNNs are useful while modelling temporal variations in the data. During the course of our literature review, we found that most of the successful neural network architectures for traffic prediction were designed using CNN and RNN units as building blocks.

An extensive summary of deep neural networks in various aspects of transportation systems is presented in Wang et al. (2019). They cover a wide range of traffic related prediction tasks using deep neural networks — traffic signal identification, traffic variables prediction, congestion identification and traffic signal control. Nguyen et al. (2018) also cover the aforementioned tasks and add three other tasks to the list, *viz.* travel demand prediction, traffic incident prediction and driver behaviour prediction. Wang et al. (2020a) surveyed the applications of deep learning in various domains which use spatio-temporal data (transportation, human mobility, crime analysis, neuroscience and location-based social networks). Their review comprises recent papers dealing with deep learning methods for tasks such as traffic variables prediction, trajectory classification, trajectory prediction, and travel mode inference. Wu et al. (2020) present a taxonomic survey of Graph Neural Networks (GNN) and highlight the applications of GNNs in different fields, including transportation. Xie et al. (2020) summarise various approaches where deep learning was used for the most common types of flows in a city — crowd flows, bike flows, and traffic flows.

Congestion prediction refers to the prediction of traffic state variables when congestion is imminent. It is a special case of traffic prediction. The relative difficulty of congestion prediction tasks in comparison to traffic prediction can be attributed to the instability of the traffic dynamics beyond the point of maximum flow (Chung, 2011). This higher relative difficulty is also obvious from the fact that the performance of a typical traffic prediction model degrades as the state of traffic approaches. The importance of deep learning for congestion prediction is also due to the relatively higher stability of deep learning models compared to other data-driven approaches (Yu et al., 2017). To the best of our knowledge, a comprehensive survey paper covering the applications of deep learning in congestion-related tasks does not exist. This survey paper is an attempt to bridge this gap in the literature. We discuss the applications of deep learning in the detection, prediction and alleviation of both types of congestion — recurring and non-recurring. The section on congestion prediction is not differentiated into recurring and non-recurring cases because the deep learning models which detect congestion from traffic images, detect both types of congestion. Whenever possible, we have attempted

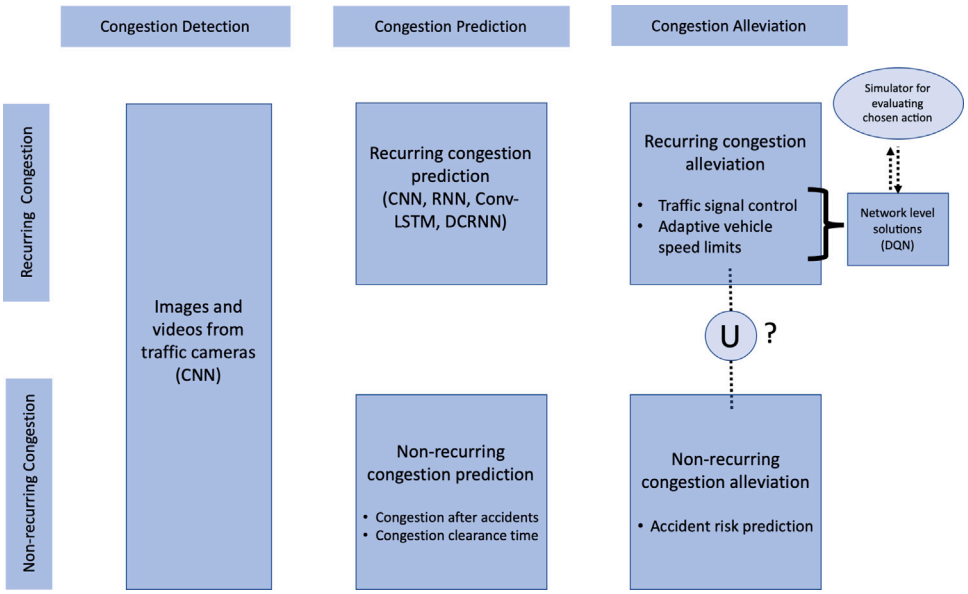

Fig. 8. Overall flowchart showing various sections of the survey. The question mark denotes the potential link between non-recurring and recurring congestion alleviation.

to incorporate the challenges from a policymaker's point of view. The inclusion of the policymaker's outlook is important in order to make research outputs deployable in a real world setting. The scope of the current survey is presented in Fig. 8.

The key design aspects of the deep learning architecture and the key aspects of the dataset used in each paper have been summarised so the reader can refer to such papers when working on similar datasets. Some papers present an extensive sensitivity analysis of their models. We have reproduced and highlighted the crucial insights (if any) in the summary presented at the end of each subsection.

Most research papers covered in this survey were published in the period 2016–2021. Sometimes, in order to briefly discuss the background of some algorithm paradigms, we have referred to classical papers from the past.

4. Deep learning for congestion detection

With increasing access to new data sources, new opportunities to automatically detect traffic congestion are being explored. Unlike the other two sections, the congestion detection models are not differentiated for recurring and non-recurring congestion. This is so because congestion detection models invariably detect both types of congestion irrespective of the causal factor behind them. They are instead differentiated on the basis of the data source used. The most commonly used sources to detect both types of congestion are images and videos obtained from traffic cameras.

The benefit of using traffic camera images is that without exception, all vehicles are captured in the traffic image. Thus, other factors like penetration ratio (percentage of vehicles being tracked) do not need to be considered. With the increasing number of cameras on roads, the cognitive load on the human identifying congestion from the images is high. In order to reduce the cognitive load, deep learning has been widely applied for the automatic detection of congestion from traffic images. Deep learning models which are known to perform well for Computer Vision (CV) tasks have been adopted to detect traffic congestion. CV refers to the task of extracting useful information from images.

Convolutional Neural Networks (CNNs) form the building block of commonly used deep learning architectures for image classification tasks. The seminal works in this field were: AlexNet (Krizhevsky et al., 2012), InceptionNet (Szegedy et al., 2015), Resnet (He et al., 2016), R-CNN (Girshick et al., 2014), Mask-RCNN (He et al., 2017), VGGNet (Simonyan and Zisserman, 2014), and YOLO (Redmon et al., 2016). Deep neural networks pre-trained on large image datasets like ImageNet (Lin et al., 2014) and COCO (Lin et al., 2014) are readily available. Three approaches are possible when using well-known architectures for traffic image classification. When the number of traffic images in the dataset available at hand is very high (order of 10000 images), these models can be trained from scratch using the available data. When the dataset available is small, the weights for the adopted deep learning model are initialised to the available pre-trained model. The third approach is to retain the pre-trained model as it is and add another module in sequence. When applied to images, the deep learning models can be used to estimate the total number of vehicles in an image, thereby allowing us to estimate the traffic density. When applied to video data (a sequence of images), the deep learning models can be used to estimate traffic speed.

Two variations of CNN based architectures (AlexNet and YOLO) are utilised (Chakraborty et al., 2018) to detect congestion using a binary classification into traffic images collected from 121 cameras from Iowa, USA over a period of 6 months. Manual

labelling of traffic images into congestion and non-congestion labels is a time-consuming task. The authors, therefore use occupancy data obtained using vehicle loop detectors (VLDs) to automatically label the images into two classes based on occupancy (20% occupancy is labelled as 'congested'). The reported accuracy for detecting congestion was 90.5% for AlexNet and 91.2% for YOLO respectively. Wang et al. (2018) compare two variations of AlexNet and VGGNet to detect congestion on traffic images obtained from more than 100 cameras from Shaanxi province, China. Their dataset is highly varied- comprising images for day and night traffic and varying weather conditions. Their results show comparable performance for both architectures (78% for AlexNet compared to 81% for VGGNet). They report that AlexNet is significantly faster to train due to the smaller size of the neural network. They use binary classification ('jam' or 'no jam'). Impedovo et al. (2019) compared the performance of YOLO and Mask-RCNN on three manually labelled datasets obtained from two traffic image data sources (GRAM and Trafficdb). The three datasets are of varying image quality — first, comprising 23435 images at low resolution (480×320p), second, comprising 7520 frames at mid resolution (640×480p), and third comprising 9390 frames at high resolution (1280×720p). They achieve congestion detection in two steps. The first step focuses on identifying the number of vehicles in each frame. In this step, the Mask-RCNN achieves an accuracy of 46%, 89%, 91% respectively while YOLO achieves an accuracy of 82%, 86%, 91% respectively. The performance of YOLO is resistant to the image quality and the training time is almost half of Mask-RCNN. They select YOLO as the object detector model and use its output as the input to the second step. In the second step, they use Resnet on the output of YOLO to predict traffic congestion as a multiclass classification task (3 classes). The reported accuracies for light, medium and heavy congestion are 99.7%, 97.2% and 95.9% respectively.

A CNN model is used in Kurniawan et al. (2018) in order to classify traffic images obtained in Jakarta, Indonesia. The data are collected for 15 days and 14 camera locations are used. They use manual labelling of the traffic images into 'jammed' and 'not jammed' classes. The reported average accuracy using 10-fold cross-validation is 89.5%. Rashmi and Shantala (2020) investigated the performance of YOLO when the traffic is highly heterogeneous. They collect one week of data from Karnataka, India. They use transfer learning with a YOLO model pre-trained on the COCO dataset. While counting vehicles in the images, YOLO performs well (accuracy between 92% and 99%) for buses, cars and motorcycles but when predicting the modes of transport which are specific to the zone of study, the performance drops below any useful level.

Summary: We observe significant differences in the image quality based on the data source. This results in differences in the model performance. The traffic images obtained from developing countries present a major challenge due to the large heterogeneity of the traffic stream. Another major difference between datasets is that when alternate sources of data are present (such as VLDs), the labels for training data can be created automatically, instead of manual labelling. If the quality of the images obtained from traffic cameras is not high, deep learning based image super-resolution can be used for improvement. Image super-resolution refers to the task of increasing the resolution of input images. Deep learning based super-resolution has been widely researched in the computer vision community, but we are yet to see its applications in improving the quality of traffic images. A comprehensive survey on deep learning applications for image super-resolution is presented in Wang et al. (2020b). Another potential avenue for improvement in congestion detection from cameras is deep learning based video frame rate increment (Jiang et al., 2018).

5. Deep learning for congestion prediction

5.1. Deep learning for recurring congestion prediction

Recurring congestion occurs due to infrastructural bottlenecks, which are insufficient to handle the peak demand. By definition, recurring congestion occurs at familiar locations in the network. In the light of this definition, the specific task while predicting recurring congestion is the prediction of daily variations in the time of occurrence and the severity of recurring congestion. The most commonly used congestion prediction task is binary classification ('jam' or 'no jam'). Some papers predict congestion as a multiclass classification task ('light', 'medium' and 'heavy' congestion). It should be noted that some papers included in this section did not predict traffic congestion but instead focused on short-term traffic prediction as a regression task (predicting speed, density, flow, queue length etc.). Such papers have been included if their model performance was reported to be stable when the state of traffic approaches congestion. For each paper listed here, we have reported the key aspects of their performance metric and the key takeaways from their sensitivity analyses. The papers discussed in this subsection are summarised in Table 1.

Using LSTM models: An LSTM model is used in Yu et al. (2017) to predict traffic speed during peak hours. They use vehicle loop detector (VLD) speed recordings from the publicly available Caltrans dataset from California, USA and attempt to predict the next hour traffic speed at each sensor location. When predicting the traffic speed during peak hours, their LSTM model is reported to achieve a MAPE of 5%. An important observation from their study is that timestamp features (encoded values for the time of day and day of the week) significantly improve the prediction performance during the peak hour. This practice of incorporating timestamps has been adopted in papers that appeared later. In this paper, each VLD is modelled separately. In the subsequent papers, researchers have developed techniques to incorporate spatial information into the LSTM models and hence propose a single model for predicting congestion at several intersections at the same time. Rahman and Hasan (2020) use an LSTM model to predict the queue lengths at intersections by incorporating the spatial information in an efficient manner. In order to predict the queue length at a query intersection for the next traffic cycle (red signal), their model takes as input the queue lengths of the query intersections and two upstream intersections at the current cycle. They then attempt to predict the queue length at the query intersection for the

Table 1
Table summarising the papers discussed in Section 5.1 (Deep learning for recurring congestion prediction).

Paper	Congestion defined on the basis of:	DNN architecture	Performance	Data source	Unique aspect
(Wang et al., 2016)	Traffic speed	erRCNN (built using CNN, RNN)	5 km/h≤RMSE≤8 km/h (horizon: 10 to 50 min)	2 ring roads Beijing (China)	Insights into congestion source detection
(Ma et al., 2015)	Traffic speed	RBM, RNN	Accuracy 88.2%	GPS data 515 road links Ningbo, China	Extensive sensitivity analysis w.r.t binary threshold on speed
(Yu et al., 2017)	Traffic speed	LSTM	MAPE: 5%	2018 VLDs (45 days) California, USA	Spatio-temporal analysis of performance
(Sun et al., 2019)	Traffic speed	CNN LSTM	90.55%≤Accuracy≤96.32% 91.89%≤Accuracy≤96.75%	2000 taxis GPS (28 days) Chengdu, China	Extensive sensitivity analysis w.r.t input horizon
(Cheng et al., 2018)	Traffic speed	Novel architecture (built using CNN, LSTM & attention)	QWK 0.52 at 60 min	349 road links (4 months) Beijing (MapBJ)	Insights into upstream and downstream flows
(Ranjan et al., 2020)	Traffic speed	Novel PredNet) (built using CNN&LSTM)	Accuracy: 84.2%	Speed heat map Seoul, S Korea	Scalable architecture
(Shin et al., 2020)	Traffic speed	LSTM	MAPE: 4.29% (urban) MAPE: 6.08% (suburban)	Urban suburban areas in & around Seoul, S. korea	Observation: variation in complexity of task based on the type of network
(Liu et al., 2017)	Traffic flow	Conv-LSTM Conv bi-dir-LSTM	MAPE 9.53 MAPE 6.98	PeMS California, USA	Bi-directional LSTM for historical data
(Li et al., 2017a)	Traffic flow	Novel DCRNN (built using encoder & decoder)	MAPE: 2.9%,3.9%,4.9% for 15, 30, 60 minutes respectively	PeMS California, USA	Incremental training using scheduled sampling; insights into effects of thresholds on load factor
(Rahman and Hasan, 2020)	Queue length	LSTM	RMSE~1	11 intersections (VLDs) 3 months Florida, USA	Efficient encoding for spatial information
(Fouladgar et al., 2017)	$\frac{average\ \ speed}{speed\ \ limit}$	CNN, RNN	–	PeMS California, USA	Node-specific thresholds for better generalisation
(Di et al., 2019)	Not applicable (pre-labelled by data provider HERE api)	Conv-LSTM	MSE:0.73 (weekdays), 0.37 (weekend)	553 road links (5 weeks) Helsinki, Finland	Congestion tree

next cycle. They use VLD data collected at 11 intersections for a period of three months in Orlando, Florida. When predicting queue lengths, they report an average RMSE close to one (as inferred from their plot).

Comparing LSTM and CNN: A comparison of the performances of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for predicting traffic congestion is presented in Sun et al. (2019). Their dataset comprises 28 days of GPS-trajectories of 2000 taxis in Chengdu, China. They used map matching to map the GPS trajectories to road segments and calculate average speed during 5-minute time slots for each segment. They then use the average speed values to define four congestion *levels* based on the average traffic speed. The reported RMSE for the average speed prediction for their best models is 3.96 *km/h*. The classification accuracy is then reported for the predicted level of congestion. They conclude that, given a sufficiently long input horizon (90 min), the performance of CNN models is as good as recurrent network models. We believe this is an important observation for two reasons. First, since the CNN models are faster to train because by construction they support GPU parallelisation in a more efficient way, it can save time on the part of researchers if CNNs are explored as a modelling option before exploring LSTMs. Second, since LSTMs are typically used to capture long-term dependencies in traffic data when used independently, they might not be very useful for short-term traffic prediction. The following papers in this subsection use combinations of CNNs and RNNs to design specialised architectures.

Using a combination of LSTM and CNN: Liu et al. (2017) present Conv-LSTM, which is composed of CNN and LSTM units. The convolution operations are used to capture the spatial dependencies. The output of convolution operations is used as inputs to the LSTM units. While using their Conv-LSTM model to predict traffic flow, they achieve a MAPE of 9.53%. Additionally, they include a

bi-directional LSTM module to include the effects of historical data and achieve a lower MAPE of 6.98%. The performance is reported to be stable across varying levels of traffic flow, hence we have included their paper as a congestion prediction model. Ranjan et al. (2020) predict city wide traffic heat maps for three prediction horizons of 10, 30, and 60 min. Since the input and output heat maps have the same dimensions, they propose a symmetric U-shaped architecture with CNN blocks at both ends (inputs and output ends). The bottleneck layer (at the highest depth) is made up of four LSTM units and skip connections are used to connect the CNN outputs at various depths. The proposed architecture is called PredNet. The dataset consists of traffic heat maps based on space–mean traffic speed collected from Seoul, S. Korea. The temporal resolution of their data is 5 min. Congestion is defined as a three-level variable based on the average speed of traffic. They report the performance of PredNet to be stable with the increasing lengths of output horizon. When predicting congestion 60 min into the future, PredNet achieves a mean accuracy of 84.2% (compared to 75.67% when Conv-LSTM is used on the same data). PredNet is also reported to be significantly faster to train (~8 times faster) compared to the Conv-LSTM model discussed previously in this paragraph.

Special focus on the heterogeneity of the road network: Some papers have highlighted the differences in the complexity of the congestion prediction task based on the heterogeneity of the road network. Such observations are particularly reported when the congestion prediction is attempted at large parts of the network for multiple time steps. Shin et al. (2020) use a three-layered LSTM network to predict the congestion levels in data collected from urban and suburban areas in and around Seoul, South Korea. The total number of road links was 1630 and the data were collected for one month at a resolution of 5 min. The dataset had 33% of the records missing. In order to handle the missing data, they propose a trend-filtering based spatio-temporal outlier detection and data correction algorithm. The model predicts traffic speeds but outputs congestion levels based on thresholds recommended by the local policy-making authority. The model performance is stable across the entire range of traffic speeds, hence we have included their work as a congestion prediction model. They report differences in the performance of the model when predicting traffic speed for two different types of road networks around Seoul (MAPE of 4.297% for suburban vs MAPE of 6.087% for urban roads). The mean absolute error (MAE) however, was higher for some suburban roads compared to urban roads (urban: 2.54 km/h, suburban: 2.78 km/h). They acknowledge that high MAE error for suburban roads, is in fact, misleading because the average speeds for suburban roads are higher. The takeaway from their paper is that while applying deep learning for predicting congestion, different types of roads present different complexity.

Cheng et al. (2018) propose a specialised architecture built using CNNs, RNNs and an attention mechanism to predict congestion levels. Their dataset, called MapBJ consists of 4 months of data collected at 349 road links in Beijing at a temporal resolution of 5 min. Each road link is labelled into one of the four congestion levels (fluency, slow, congestion, extreme congestion), based on a speed-limit normalised variable called 'limit level'. The exact mathematical representation of the 'limit level' is not presented, however, the idea is similar to using a ratio of the actual speed to the speed limit, given by ($\frac{average\ speed}{speed\ limit}$). The road network is converted to a representative graph with road links being represented as vertices and intersections as edges. The upstream and downstream vertices of the target vertex are grouped by vertex order. The input traffic conditions from different groups of vertices are used as inputs to the CNN module, followed by the RNN module. Thereafter, an attention model is used to assign different weights to different groups of vertices. The separate modules for upstream and downstream roads allow insights into the effects of upstream and downstream links on the congestion prediction at the target link. While predicting congestion state 60 min into the future, *higher* weights are observed for higher-order downstream neighbours. On the other hand, *lower* weights are observed for higher-order upstream neighbours. Such analyses provide useful insights into the demand and the flow of traffic. They use a metric called *quadratic weighted Kappa (QWK)* (Ben-David, 2008). They report an average QWK of around 0.6. They report QWK values to be 0.69, 0.63, 0.57 and 0.52 for predicting at 15, 30, 45 and 60 min respectively. For comparison, the QWK for a stacked autoencoder model was 0.68, 0.62, 0.56 and 0.49 for predicting at 15, 30, 45 and 60 min respectively.

Large-scale recurring congestion prediction (Congestion propagation) Congestion propagation can be understood as a special case of congestion prediction. It is the study of the evolution of congestion in a larger part of the network than what is usually covered by congestion prediction models. Congestion propagation can be studied under the same heading as congestion prediction. However, during the literature search for this survey, we observed that special challenges are encountered when deep learning models are used to predict traffic congestion for the entire network.

A specialised architecture called DCRNN was proposed in Li et al. (2017a) in order to predict traffic flow for several time steps (15 min, 30 min and 1 h). The dataset used is PeMS. The model performance is unchanged during peak hours and under varying levels of flow, hence this paper has been included as a congestion prediction model. DCRNN consists of an encoder and a decoder component. The encoder takes traffic flow data with spatial parameters encoded into a graph and outputs *hidden states*. The decoder attempts to predict the next-step traffic flow, either using the *hidden states* from the encoder with a probability ρ, or using the ground truth data with a probability $(1-\rho)$. At the start of the training, the value of ρ is close to 1 and is decreased to 0 by the end of the training. The reported MAPE are 2.9%, 3.9% and 4.9% for 15, 30 and 60 min respectively. Andreoletti et al. (2019) also used a DCRNN model to predict congestion as a binary classification task based on traffic density. The threshold for binary classification is defined using road link specific load factors (α). The load factor (α) for a link is defined as the ratio of current traffic density to that of the average traffic density ($\alpha = \frac{current\ traffic\ density}{average\ traffic\ density}$). The reported congestion prediction accuracy is 96.67% when α is set to 3. At high values of α, the rate of false negatives (FN) increases. A high value of alpha implies that only very high traffic density is classified as congestion. A value of $\alpha = 3$ appears to be optimum with a false negative rate of 2.4%. The optimal choice of threshold in order to achieve better generalisation of deep learning models has been explored further in papers that appeared later on.

Ma et al. (2015) used a Restricted Boltzmann Machine (RBM) combined with a recurrent neural network (RNN) model to predict the evolution of congestion. The dataset comprises GPS-trajectories of taxis plying on 515 road links in Ningbo, China. They use a

network-wide threshold on traffic speed to determine whether the predicted traffic speed implies congestion. They report an average accuracy of 88.2%. An interesting observation from their sensitivity analysis is that an increase in the threshold degrades the model performance. They hypothesise that this might be due to the higher fluctuations in the congestion propagation patterns when a higher percentage of road links fall into the congested category. Fouladgar et al. (2017) used the PeMS dataset from California and proposed a distributed network where each intersection of the road network was modelled using a separate deep learning model. They used a combination of CNN and RNN architectures to predict the congestion levels. In order to binarise congestion, they introduced *node-specific* thresholds, instead of a *network-wide* universal threshold, thereby giving their model more expressive power compared to (Ma et al., 2015). The node-specific thresholds are defined using the ratio of the average speed to the speed limit ($threshold = \frac{average\ speed}{speed\ limit}$). The sensitivity analysis of their model revealed a *drop* in the model's performance as the congestion levels increase. They hypothesise that this drop is due to class imbalance (high number of data points for non-congested cases). They attempted to remedy this *drop* by preferentially weighing the data points where congestion was high. The preferential weighting is achieved by modifying their mean squared error loss function to include a penalty ($w_i * \beta_i$) for each data point i. The difference between the predicted flow and the ground truth is β_i. The value of w_i is equal to 1 if the *traffic flow* for data point i is less than $0.5 * (max\ flow)$ and 0 otherwise. The β value is a measure of the prediction error. Now, using the variable w_i, the model imposes an extra penalty for errors when the traffic flow was more than half the maximum flow.

Wang et al. (2016) proposed erRCNN, which is built using CNN units followed by RNN units. In Fig. 7, we observed that when the flow exceeds beyond a threshold, the speed–density curve becomes scattered and abrupt changes in the average speed are observed. Their erRCNN architecture is shown to handle these abrupt changes. The error-correcting RNN allows the model to be updated when the prediction performance drops due to a change in the state of traffic. Thus, the model is capable of handling streaming data. They used a GPS dataset collected from 2 major ring roads in Beijing, China. The reported RMSE for speed prediction varies between 5 km/h when the prediction horizon is 10 min and 8 km/h when the prediction horizon is 50 min. Additionally, in order to understand the sources of congestion propagation, they propose a metric called segment importance. Each segment influences the traffic on other segments. If the road in question has n segments, it can be assumed that the trained errRCNN model has learnt a mapping between traffic speeds at each segment at time t and the traffic speeds at each segment at time $t+1$. Mathematically, $erRCNN\ ([([v^t_{seg_1}, v^t_{seg_2}..v^t_{seg_n}])]) = [v^{t+1}_{seg_1}, v^{t+1}_{seg_2}..v^{t+1}_{seg_n}]$. Using this assumption, first they define the influence of seg_i on seg_j as the derivative of $v^{t+1}_{seg_j}$ w.r.t $v^t_{seg_i}$. Finally, they define the segment importance of a segment as the sum of the influences to all other segments. They then map the segments with high segment importance to physical locations in the network and uncover some sources of congestion (such as an intersection connecting two highways). Even though these observations are intuitive, their contribution is significant because it throws light on how deep learning models can be used to provide interpretable insights, thereby promoting wider acceptance by transport authorities.

Deep learning for congestion trees: Another popular method that has been historically used to model the propagation of congestion utilises the concept of congestion trees. Attempts have been made to model the evolution of congestion trees using deep learning. A congestion tree is formed when congestion on one road segment results in the building up of congestion onto the adjacent road segment. Several congestion trees can be combined by removing the redundancy between them. This gives rise to a congestion graph. Di et al. (2019) remove the redundancy between the congestion trees by creating a directed acyclic graph (DAG) through combination of the congestion trees. This DAG is then converted into a spatial matrix of congestion levels, with each cell of the matrix representing at most one segment. The spatial matrix helps preserve the adjacency information between the road segments. A sequence of these spatial matrices (SM) is then passed to a Conv-LSTM model for predicting the SM at the next time step. The predicted SM is transformed back to the congestion graph and then to the congestion tree, which can then be used to provide a visual representation of the predicted evolution of congestion. When using a 5-minute prediction horizon, they reported the Mean Squared Error (MSE) of 0.27 for weekdays and 0.07 for weekend traffic. MSE for 15 min was 0.73 (weekday) and 0.37 (weekend). In their result, MSE has no units because it is not computed for a traffic variable, but for the spatial matrices. For comparison, the LSTM model achieved an MSE of 0.59 for weekdays and 0.32 for weekends when using a 5-minute prediction horizon.

Summary: In this subsection, various aspects of deep learning applications in predicting recurring congestion are discussed. The takeaways from this subsection are summarised below:

- We observe that there have been very few attempts to comprehensively compare the performance of different deep learning models while keeping the dataset and the specific prediction task fixed. We know that the performance of a deep learning model is largely dependent on the choice of hyperparameters (such as the number of hidden layers, number of convolutions, learning rate etc.). When new deep learning architectures are presented for specific tasks, a fair comparison with previously used architectures might not be feasible. The reason behind this is that the best performance from a deep learning model involves training a large number of models to determine the best set of hyperparameters. So, it is not plausible to do such fine tuning for all the previous work.
  The solution might be to establish public benchmarks for each dataset while keeping the test, train, validation split fixed. Such dataset specific benchmarks are widely popular in the computer vision community. They might significantly reduce the duplication of efforts to reproduce results from a previous paper. Another benefit of such benchmarks would be to reduce the duplication of data pre-processing as it can be a more time consuming task than training a deep neural network itself.
- Particularly, while predicting congestion over a large part of the network for several time steps, we observe that extensive sensitivity analyses has been reported in some papers in order to reveal the temporal and spatial variation in performance. Highlighting such differences in performance has resulted in future work being targeted at improving the spatial and temporal

generalisation of the deep learning models. We believe this is a good trend and future research should include more of such analyses. It also helps us to understand the limits of short-term traffic prediction using deep learning. Based on the papers presented here, the maximum time horizon of prediction appears to be 60 min.
We believe that future attempts to increase the prediction horizon will be useful to garner trust in the deep learning based solutions for traffic congestion prediction.

- We observe a dearth of papers which present deep learning models that are updated as new data arrives. This is known as online learning. Online learning is a framework and not a model. So, theoretically any deep learning model can be integrated into an online learning framework and model updates can be demonstrated. So far, it is not popular because live traffic data are not easily available to researchers.
  Even when streaming new traffic data are not available, online learning capabilities can still be demonstrated by using temporal splits of the historical data and evaluate model performance as it gets trained on increasing amounts of data.

5.2. Deep learning for non-recurring congestion

An exhaustive list of causes behind non-recurring congestion is not known (McGroarty, 2010). New causes are added as new data sources become available and new correlations and causalities are established. Some well investigated causes of non-recurring congestion are traffic accidents, varying weather conditions, disasters and planned events. In this survey, we focus on deep learning applications for predicting congestion due to accidents. The reasons for this focus on accidents are threefold. First, among this list of well-studied causes, traffic accidents are the leading cause behind a large percentage of non-recurring congestion (Hallenbeck et al., 2003). Second, deep learning has been widely used to predict congestion after traffic accidents. Third, investigating traffic congestion due to other causative factors (weather, planned events and disasters) are best suited for scenario based studies using traffic simulators. Such studies are usually conducted in the planning stage and hence the high computation time of using traffic simulators do not present a challenge (Aljamal et al., 2018). A literature search reveals that there is a wide variation in the specific deep learning task when using deep learning methods to predict congestion after traffic accidents. We have grouped the research into the following two clusters:

- prediction of post accident traffic congestion
- prediction of post accident congestion clearance time

The papers discussed in this subsection are summarised in Table 2.

Post accident traffic congestion prediction: Sun et al. (2017) propose a CNN based architecture to predict traffic flow after accidents. The traffic speed data obtained from the traffic information system (HEREapi) is converted to traffic heat map images. If the speed (v) is less than 80 miles/h, the pixel value p is set to $(1 - v/80) * 255$ and 0 otherwise. A single threshold is used to define a binary classification task with two classes — recurring and non-recurring congestion. The reported accuracy is 86.6% with a false positive rate of 13.71% and a false negative rate of 4.44% (FN: model wrongly classifies non-recurring congestion as recurring congestion). An interesting contribution from their work is the use of crossover operator for reducing data imbalance issues (low number of 'accident' data points compared to 'no accident' data points). Crossover is a technique commonly used in genetics to model the creation of new chromosomes by partial exchanges of the genetic material of parent chromosomes. They hypothesise that various traffic data points collected within a short time range have the same event label and hence, applying crossover to random locations in the traffic heat maps results in data augmentation without compromising the data quality.

Yu et al. (2017) propose a mixture model which has two components, one composed of LSTM and the other composed of an autoencoder. The incident data are fed to the autoencoder and the traffic data are fed to the LSTM. Finally, the outputs from the two components are concatenated and a fully connected layer is used to output traffic speed at each sensor location. They use vehicle loop detector (VLD) speed recordings from the publicly available Caltrans dataset from California, USA. While predicting post incident traffic speed for a prediction horizon of 3 h, their model achieves a MAPE of 0.97%. For comparison, an LSTM model achieves a MAPE of 1.00% and a three layered feed forward neural network achieves 3.65%. An interesting contribution of their work is the use of signal stimulation to investigate the effects of abrupt reductions of input speed on the model response. They report that the model's response remains unchanged when the stimulations last only for short durations (<5 min), thereby suggesting that the model is robust to the minor fluctuations in the input data. Additionally, they report that the model response is amplified when the stimulations are injected during peak hours.

Fukuda et al. (2020) propose X-DCRNN, which is an extension of DCRNN in order to input the incident data explicitly. Their dataset is created using simulations on the microscopic traffic simulator MATES (Yoshimura, 2006). The traffic demand was calibrated using the meta data obtained from the local transportation authority. The network in the simulator is based on the central business district of Okayama city in Japan, consisting of 206 traffic sensors across 339 road segments and spread across 3 square kilometres. Their model predicts post incidence traffic speeds and the errors are reported for incident segment and the corresponding downstream segments. For the incident road segment, the reported MAE is 0.74 miles/h and the RMSE is 0.87 miles/h. For comparison, on the same dataset, the DCRNN model achieved a MAE of 1.97 miles/h and an RMSE of 5.64 miles/h. While predicting the traffic on the immediate downstream road segment, both DCRNN and X-DCRNN achieve a similar level of performance. For immediate downstream segment, X-DCRNN achieves MAE of 3.68 miles/h and RMSE of 6.39 miles/h. For comparison, the DCRNN model achieved MAE of 3.81 miles/h and RMSE of 6.33 miles/h.

Predicting congestion clearance time: Congestion clearance time is a useful indicator of assessing the impact of accidents on traffic congestion. After an accident, traffic flow reduces due to restricted movement on affected lanes. Typically, the traffic flow

Table 2
Table summarising the papers discussed in Section 5.2 (Deep learning for non-recurring congestion prediction).

Paper	DNN architecture	Performance	Data source	Unique aspect
(Sun et al., 2017)	CNN	Accuracy = 86.6% FPR = 13.71% FNR = 4.44%	Traffic speed data from HEREapi (6 days: train, 1 day: validation)	Data augmentation using crossover
(Yu et al., 2017)	Mixture model (LSTM and autoencoder)	MAPE = 0.97% (predicting post accident traffic speed)	2018 VLDs (45 days) California, USA	Robust model tested with stimulation response
(Fukuda et al., 2020)	X-DCRNN extension of DCRNN (Li et al., 2017a)	MAE = 3.68 miles/h RMSE = 6.39 miles/h (predicting post accident traffic speed)	Simulated data for 206 traffic sensors, 339 road segments generated using MATES (calibrated demand from Okayama, Japan)	Large number of training data points possible due to simulated data
(Zhang et al., 2019a)	FFNN (2 hidden layers)	MAPE = 40% RMSE = 8.3 min	173 VLDs from Shanghai 4017 accident records from police records	Performance comparable to multilinear regression suggesting the use of other ANN architectures
(Lin and Li, 2020)	FFNN (one hidden layer)	MAPE = 9.33% RMSE = 7.62 min	Anonymous Navigation System Beijing, China	Novel data fusion and model update with new data
(Li et al., 2020b)	Specialised fusionRBM	MAPE = 20.23% RMSE = 11.84 min.	968 accidents from HSIS& traffic data from PeMS dataset California, USA	Specialised architecture for data fusion

eventually decreases to a minimum value and then recovers. When modelling congestion clearance time, most papers presented here focus on the time duration between the instant when the maximum level of congestion is reached (lowest flow) to the time instant when the flow returns back to pre-accident levels. A FFNN with two hidden layers is used in Zhang et al. (2019a). They use average speed obtained from 173 vehicle loop detectors (VLDs) in Shanghai, China and accident data from traffic police records to create input vectors having 9 features. While predicting congestion clearance time, their model achieves a MAPE of 40% and an RMSE of 8.3 min. For comparison, a multilinear regression model achieved a MAPE of 49.8% and an RMSE of 10.22 min. The total number of accident records in the dataset was 4017.

Lin and Li (2020) leverage the idea that in the instances of post accident congestion, the exact details about the accident are not available instantly. More information about the type, location, severity and affected lanes are available as the damage is assessed by the bystanders, involved parties or emergency response. They propose a framework capable of updating the prediction with the arrival of new information. They define non-recurring congestion prediction as a multiclass classification task (5 classes). The class of congestion is defined using thresholds on the maximum value of congestion delay index ($CDI = \frac{v_{free}}{v}$) reached after the accident. The congestion clearance time is defined as the time elapsed while the CDI returns from its maximum value to pre-accident levels. They then predict the congestion clearance time for each type of accident. They propose a framework, consisting of FFNN with one hidden layer that is capable of updating the model using the new data available during the course of congestion clearance time. The model update feature results in a significant performance improvement for the most severe accident class: RMSE (minutes) decreases from 10.8 to 7.62 and MAPE decreases from 17.4% to 9.33%. Their data was collected using an anonymous navigation system from Beijing, China.

Li et al. (2020b) propose an architecture called fusion RBM which is created by concatenating the outputs of two sets of stacked RBMs. The fusion aspect of their model is inspired by the fact that the accident data consist of categorical variables while the traffic data are continuous. One stacked RBM unit takes the categorical accident data as input while the other stacked RBM unit takes the continuous data as input. Finally, the outputs from both the units are concatenated and passed through a single neuron to finally output the congestion clearance time. The predicted congestion clearance time is quantified into ten levels using 10 min increments (0–10, 10–20 and so on). The reported MAPE is 20.23% and the RMSE is 11.84 min. They used traffic data collected from California (PeMS) and traffic incidents data collected from a highway safety information system (HSIS). The total number of accidents in their dataset was 968 with a mean congestion clearing time of 37 min.

Summary: We discussed the applications of deep learning in predicting post accident traffic congestion and its clearance time. The key difference from recurring congestion prediction was that data from multiple sources must be fused in order to predict post accident congestion. Each paper presented here handled the challenge of data fusion in different ways, with almost no consensus between them. Benchmarking of standard data fusion algorithms from multiple sources might be helpful to provide insights to future researchers about the most efficient techniques for traffic data fusion. Data fusion has been extensively studied in the Internet of things (IOT) community. IOT research is focused on pervasive communication between different devices, hence efficient data fusion has been researched extensively. Transportation researchers can draw inspiration from such sources and explore the possibilities of improving post accident congestion prediction using efficient data fusion techniques. Gao et al. (2020) compare the data fusion performance of different deep learning architectures. Most of the architectures presented in their survey are commonly used for deep learning based congestion prediction task, hence it would be interesting to use the insights presented therein. Predicting post accident traffic congestion can also be understood in the light of the traffic hysteresis curve presented in Section 2.2.

6. Deep learning for congestion alleviation

The congestion alleviation techniques differ significantly for recurring and non-recurring congestion. On the one hand, recurring congestion is caused due to infrastructural bottlenecks which are insufficient to handle the peak demand of traffic. So, the deep learning solutions for recurring congestion are targeted at decreasing the severity of the recurring congestion by distributing the demand in an optimal fashion. On the other hand, non-recurring congestion is caused primarily due to accidents. So, the deep learning applications for alleviating non-recurring congestion are targeted at reducing accidents. Deep learning has been widely used to predict the accident risk. The predicted accident risk can be used to alert drivers or to impose speed restrictions in order to reduce accidents. At the end of this section, we discuss the potential connection between the efforts to reduce recurring congestion and the efforts to reduce non-recurring congestion. As discussed in the previous Section 5.1, other causes of non-recurring congestion such as planned events, bad weather and natural disasters are best suited for scenario based studies using traffic simulators and hence, deep learning has not been applied extensively for non-recurring congestion due to those causes.

6.1. Deep learning for recurring congestion alleviation

As soon as recurring congestion is predicted by a congestion prediction model, alleviation measures can be put in place to reduce the build up of congestion. At the network level, this can be achieved by controlling the supply parameters such as traffic signal control at intersections, ramp metering, imposing speed limit restrictions and imposing lane use restrictions. At the individual level, this can be achieved by using descriptive, prescriptive or hybrid methods. Descriptive methods involve broadcasting vital information about the network traffic conditions to the drivers and helping them make an informed decision about their trip start time and chosen route. Prescriptive methods involve trip specific suggestions to drivers (such as optimal start time and optimal route choice). Hybrid methods use a combination of descriptive and prescriptive methods. The descriptive and prescriptive methods face significant challenges due to the additional layer of human behaviour modelling in those approaches. In this survey, we focus on the recent research using deep learning for congestion alleviation at the network level.

Challenges in using demand side solutions: The challenges encountered when using descriptive and prescriptive methods for congestion alleviation are presented in Balakrishna et al. (2013). The biggest challenge is the information flow back from the drivers to the network. The information is crucial in order to estimate the effect of the suggested route on the roads. They strongly argue in favour of the need for large-scale tracking data. Due to the high penetration of smart phones in urban areas, such real time traffic information is increasingly being relayed to the drivers through their mobile phones. The challenge however arises from the fact that the descriptive and prescriptive solutions are often grouped together. When a user queries for driving directions, the preferred routes returned by the navigation application incorporates some route suggestions. The final goal of such personalised route suggestions is not known, since the companies which provide such services present mobility as a service and the algorithmic details are trade secrets. A survey conducted in 2018 in the US reports that 87% of drivers use some navigational application to get suggestions for driving directions (Manifest, 2018). A report released by Google Maps states that more than a billion kilometres of travel per day are tracked using their application (Google, 2020). When the coverage and compliance of such personalised route suggestions increase, it can potentially alter the user equilibrium in the transportation system. Since such efforts from the private sector are targeted at creating route choice as a service and the exact methods are black boxed as trade secrets, it is difficult to estimate the potential drawbacks related to the equity of such solutions. In the absence of real data, researchers have used economic instruments as an attempt to increase the compliance and hence, ease the task of modelling the human behaviour. The compliance of advisory measures can be increased using punitive measures or rewards. The choice between which of the two approaches proves to be more effective, is a difficult one to make. The effectiveness of punitive measures is similar across various geographical locations whereas the effectiveness of reward based measures has been found to vary significantly (Tillema et al., 2013; Li et al., 2019). Two commonly used punitive measures are congestion tolling and tradable tokens. de Palma et al. (2018) present a methodology for comparing the effects of both approaches and strongly argue in favour of tradable tokens. Various ways in which new and emerging technology can be used to achieve efficient congestion tolling are discussed in de Palma and Lindsey (2011). One of their suggestions is automatic number plate recognition (ANPR). Deep learning has been widely applied for ANPR. However, this application is not covered in this survey. The interested reader is referred to Connie et al. (2018), Khan and Ullah (2019). The challenges encountered in applying ANPR to developing countries are presented in Kyaw et al. (2018).

Supply side solutions at the network level: We focus on the deep learning applications for congestion alleviation at the network level. The majority of such efforts use deep reinforcement learning. As discussed in Section 2.1, a reinforcement learning framework has four main components: agent, action, reward and the environment. The concept of *policy* fits well into the traffic signal control problem. The best policy (*e.g.*, order of the red and green lights in different directions at an intersection) can be found by optimising the model for which the reward (*e.g.* maximum cumulative flow in the network) is minimised. In the recent research, deep Q-network (DQN) is being commonly used and authors are increasingly interested in investigating the equity aspect of such solutions. The traffic controlling infrastructure (such as traffic signal or speed limit signs) is modelled as the agent. The choice of agents determines which actions are allowed. The papers are arranged into two headings based on the chosen agent.

- Adaptive Traffic Signal Control (TSC)
- Variable Speed Limit Control (VSLC)

The papers discussed in this subsection are summarised in Table 3

Table 3
Summary of papers discussed in Section 6.1 (Supply side solutions using deep learning for recurring congestion alleviation).

Paper	DNN architecture	Reward function	Performance	Data source	Unique aspect
(Genders and Razavi, 2016)	CNN based DQN	Cumulative delay queue length	Cumulative delay = 719 seconds queue length = 13	Small network (one intersection with 4 segments each with 4 lanes)	Identified several important challenges (such as fairness)
(Yen et al., 2020)	CNN based DSARSA	Power metric = $\frac{maximum\ throughput}{end-to-end\ delay}$	Power metric = 35	9 intersections (3X3 grid)	Novel reward function DQN convergence challenges highlighted
(Zeng et al., 2018)	LSTM based DQN	Average waiting time	Average waiting time = 17.71 s	Small network (one intersection with 4 segments)	Performance stable at low penetration ratios <50%
(Genders and Razavi, 2019)	FFNN based DQN	End-to-end-delay	–	Small network (one intersection with 4 segments)	40% improvement over actuated signal control
Dusparic et al. (2016)	(Classical RL)	–	–	–	Priority vehicles considered
(Shabestray and Abdulhai, 2019)	CNN based RL	Cumulative delay at intersections	For transit only average TT :292 s compared to 343 s for actuated control	6 intersections Ontario, Canada (calibrated demand)	1. Multi-modal signal controller 2. Calibrated demand 3. Head-count based priority
(Wu et al., 2020b)	LSTM based DQN	Several components including queue lengths, mean vehicle delay, number of pedestrians delayed	During peak hours 50% reduction in queue length compared to other baselines	Multiple intersections simulator: SUMO	Pedestrian waiting incorporated into reward function
(Gregurić et al., 2020)	CNN based DQN	Three components 1. Traffic flow 2. Safety constraints 3. Driver comfort	13.5% increase in average speed 25% decrease in density compared to no vehicle speed limit (VSL case)	8 km of highway simulator: VISSIM	Novel reward function
(Wu et al., 2020a)	FFNN based DRL	Four components 1. Increasing flow 2. Decreasing travel time 3. Reducing rapid acceleration 4. Reducing emissions	5.8% decrease in average travel time compared to no vehicle speed limit (VSL case)	850 m of highway simulator: SUMO	Assuming autonomous vehicles and allows for abrupt changes in speed limit and lane specific limits

6.1.1. Adaptive traffic signal control (TSC)

A CNN based DQN was presented in Genders and Razavi (2016) in order to model the signal controlling agents in a traffic simulator. The goal was to maximise the *network-wide* throughput by optimising the behaviour of these agents. They analyse the performance of their model based on three parameters — the cumulative delay, the queue length at intersections, and the average travel time. They model one intersection with four segments from four directions each having four lanes. The vehicles are loaded at varying flow rates between 0 to 450 vehicles/h. The left and right turning traffic is loaded using the inverse Weibull distribution (Rinne, 2008) while the through traffic is loaded using the Burr distribution (Tadikamalla, 1980). Using CNN based DQN to control the actions of the traffic controlling agent, they achieve a cumulative delay of 719 s and an average queue length of 13 vehicles. For comparison, a DQN based on FFNN with one hidden layer achieves an average queue length of 33. They highlight the need to study the *fairness* aspect of the policy optimised by the DQN. In the absence of a fairness metric of the model, it is possible that the algorithm favours or disfavours specific movements of traffic. They assert that the fairest policy might not result in the best throughput of traffic — hence, there is a need to search for a balance between the two objectives by tweaking the reward function. Yen et al. (2020) attempt to address this difference of objectives by incorporating a *power metric* in their reward function. The power metric is defined as the ratio of maximum throughput to the end-to-end delay ($\frac{flow}{time}$). When the power metric is maximised, the throughput (the flow of traffic) is maximised and the end-to-end delay (travel time) is minimised. Their network consists of nine intersections (3×3 grid). Using power metric as the reward function however did not result in convergence of the DQN models. Hence, they used DSARSA instead of DQN to achieve convergence. SARSA stands for State–Action–Reward–State–Action. It was introduced in Rummery and Niranjan (1994) and differs from DQN in the choice of the next action. The SARSA

algorithm uses the same policy to choose the next action as well as update the reward function whereas the DQN uses a different (greedy) policy to choose the next action. Using their CNN based DSARSA, they achieved a power metric of 35.

Another attempt in this direction is Zeng et al. (2018) who propose DRQN (Deep Recurrent Q Learning) using LSTM based DQN. They report an average waiting time of 17.71 s when using DRQN on their synthetic network consisting of a single four-way intersection. The distribution used for loading the vehicles is binomial. The performance of LSTM based DQN is similar to that of a CNN based DQN as used in Genders and Razavi (2016). However, the performance of the LSTM-DQN model is stable when the penetration ratio is varied. Penetration ratio is defined as the percentage of vehicles on the road who shared their data. The CNN-based models perform poorly when the penetration ratio falls below 0.5. At a penetration ratio of 1 (implying data from all vehicles are available), the performance is similar for both models. Genders and Razavi (2019) use a feed forward neural network (FFNN) based DQN to optimise the traffic control policy during peak hours. Using their synthetic network comprising a single intersection, they report a reduction of around 40% in the average total vehicle delay when compared to the traditionally used actuated method of traffic signal control (Newell, 1989).

Most of the research discussed in this paragraph uses a traffic simulator to observe the effect of change in traffic-controlling policy at each intersection. We observed that SUMO (Behrisch et al., 2011) is the most commonly used traffic simulator for this purpose. The main reason behind the popularity of SUMO is that it has a python interface. Python is the most commonly used language for deep learning research, hence the integration with SUMO is smooth.

Fairness is an important factor for deep learning based solutions for TSCs in order to ensure wide acceptance of such solutions by the transport-planners and policymakers. When priority vehicles are taken into account, the importance of fairness increases significantly. Priority vehicles include transit vehicles like buses and emergency transportation like ambulances. Researchers have tried to incorporate fairness constraints into the reward function of reinforcement learning solutions. Dusparic et al. (2016) used a classical (not using neural network) reinforcement learning algorithm and modified their reward function so as to penalise the signal controlling agent each time a priority vehicle is queuing at an intersection. Similar penalties have been adopted by the deep learning community in order to demonstrate fairness in the deep learning based reinforcement learning solutions. Shabestray and Abdulhai (2019) propose a multi-modal signal controller using DQN and analyse the effects on various modes of transportation. Unlike other studies summarised in this subsection, this is the only paper which used real data to calibrate the traffic demand. They modelled a road network with 6 intersections representing a busy part of a city in Ontario, Canada using the commercially available traffic simulator PARAMICS. Instead of prioritising vehicles specifically, they use a person-based performance of the controller in their reward function. This automatically prioritises vehicles with higher head-count, implying that a bus has a higher priority than a car; a fully-occupied bus has a higher priority than a partially-occupied bus. The downside is that high-priority but low head-count vehicles like ambulances are at a disadvantage when this approach is used. Wu et al. (2020b) used a synthetic network with multiple intersections modelled in SUMO and proposed an LSTM based DQN to assign priority to vehicles and also include the pedestrian waiting time in their reward function. This concept of using power metric in the reward function is inspired by the literature on wireless networks (Kleinrock, 2018).

6.1.2. Variable Speed Limit Control (VSLC)

Similar to TSC, the VSLC methods are targeted at reducing the severity of an impending traffic congestion. The specific objective is to maximise the total flow in the network while minimising the average travel time. The design of the reinforcement learning framework for VLSC is similar to that of TSC. Here, the VSLC signs are modelled as the action taking agents. The actions allowed by the VSLC agent are to post variable speed limit signs on the variable message sign board (VMS). A microscopic traffic simulator is used to evaluate the effects of the action taken by the VSLC agent. However, TSC and VSLC differ significantly in their potential to be implemented in the real world. The challenge in the real-world implementation of VSLC stems from the potential lack of compliance on the part of drivers. Hence, in order to ensure higher compliance, driver comfort must be taken into account when formulating VSLC based solutions. In comparison, driver compliance is less of a concern in case of TSC based solutions because in practice, the instances of traffic light violation are far lower than traffic speed violations (Wu and Hsu, 2021). The research summarised here has attempted to incorporate driver comfort metrics when using deep reinforcement learning models to find the optimal VSLC.

A three layered CNN based DQN model is presented in Gregurić et al. (2020). Their reward function is a weighted sum of three components. The first component is aimed at increasing traffic flow. The second component ensures safety by imposing constraints on the magnitude of variations in VMS posted speed limits. The third component ensures driving comfort by minimising the oscillations in the VMS posted speed limits. Their network was modelled using VISSIM microscopic traffic simulator and consists of eight kilometres of highway with one variable message sign board (VMS). The synthetic demand was sufficient to create 20 min of simulated congestion on the network. Using their DQN model, they demonstrate a 13.5% increase in the average speed and 25% decrease in the average traffic density. Wu et al. (2020a) consider a futuristic scenario where the connected and autonomous vehicles are ubiquitous. In such a scenario, the high compliance factor allows the VSLC agent to make more dynamic changes. They present a differential vehicle speed limit model, by applying lane specific variable speed limits. They train four different models for different rewards: increasing flow, decreasing travel time, reducing rapid acceleration and reducing emissions. The traffic demand used in this study is more realistic compared to previous work. The network is a representation of 850 metres of a major highway from California and the demand used was calibrated using the data available for the highway. The mode composition was 85% passenger vehicles and 15% buses and trucks. Compared to the no-VSL case, their DVSL model achieves a 5.8% reduction in average travel time. They used SUMO as their microscopic traffic simulator and include specific details about how different functionalities in the SUMO application interface can be utilised to implement various components of the reward function.

Summary: We discussed the applications of deep learning when applied to limit supply using network level control of traffic. We observe two major challenges in the current state of research in this field as summarised below:

- The small size of road network and the homogeneity of traffic stream used for these studies poses a significant challenge to the validation of the research as a deployment option. The traffic demand is created using mathematical distributions rather than using the real traffic demand. Only one of the papers presented here used real data to calibrate their demand. Secondly, the reported performance of the DQN models is often not compared with the existing methods of network control (such as fixed phase transitions, Actuated). Instead, the comparison between DQNs using different neural network architectures is common. Unless the DQN based solutions are demonstrated to perform better than the existing methods, the real-world deployment of such solutions might be difficult.
- Another aspect discussed here was the lack of deep learning based solutions for managing the travel demand. The majority of large-scale solutions are provided by the private sector and the academic research is limited by data availability. Personal mobility data when fused with other sources, can reveal a lot of information about the travellers, hence the hesitation of the travellers public in sharing data is justified. We believe that a wide adoption of secure data pipelines in the research community might be an important step in garnering trust among the users who share data. Privacy preserving data fusion techniques have been explored in the Internet of things (IOT) community. A summary is presented in Ding et al. (2019). Transportation researchers can draw inspiration from such sources and adopt the best practices for privacy preservation.

6.2. Deep learning for non-recurring congestion alleviation

Accidents are the leading cause of non-recurring congestion. Hence an accurate prediction of the short-term accident risk enables us to proactively alert drivers about the same and potentially instruct them to reduce speed (Ren et al., 2018). Recently, deep learning has been widely used to predict accident risk. In order to proceed with accident risk prediction, the common practice is to divide the study area into grids and then predict the number of accidents in each grid at different time intervals. The accident risk prediction can be designed as a binary prediction task where the deep learning model is trained to predict a label ('high risk' or 'low risk') for each grid. It can also be designed as a regression task where the deep learning model is trained to predict the potential number of accidents in the grid. The papers discussed in this subsection are summarised in Table 4.

Demonstrating transfer learning capabilities Najjar et al. (2017) use satellite imagery as inputs to their model. They use a model similar to AlexNet (Fig. 2) in order to predict the accident risk into three levels. Gupta et al. (2019), Najjar et al. (2017) have demonstrated transfer learning capabilities for the task of accident risk prediction. They trained their deep learning models using the data from one city and used the trained model to predict the accident risk in a different city (New York → Denver; London → New York). Gupta et al. (2019) report accuracies of 74.77% and 76.20% when using the model, which was trained on London data, to predict on data from New York and Denver respectively. Similarly, Najjar et al. (2017) trained their model on data from New York and obtained an accuracy of 78.2%. They then report comparable performance (73.1%) when using the model from New York to make predictions on data from Denver.

Spatial correlations and heterogeneity of the road networks: The road network characteristics, the drivers' behaviour and the travel demand vary spatially. On the one hand, this spatial heterogeneity results in a spatial variation in the prediction performance of deep learning models. On the other hand, it presents opportunities to leverage on the high correlation between data collected from sensors which were located in each other's vicinity.

In Zhao et al. (2018), the authors apply principal component analysis (PCA) on the traffic and accidents data and then use a feed forward neural network with two hidden layers to predict the risk of a crash. Their data are based on one year of car accident records in the UK. PCA is used to identify un-correlated components in a dataset. The authors advocate the use of PCA due to the fact that in a transportation data set, many associated sensors and vehicles produce data which are correlated to some extent. Their experimental results show a small increase (~1%) of accuracy in crash risk prediction when using PCA compared to when not using it.

García de Soto et al. (2018) used a FFNN with one hidden layer to predict the accident risk. They defined three levels of accident severity and predict the risk for each of the three levels. Their dataset is based on 2 years of traffic and incident data from 13109 road segments in Switzerland. The data from each accident were encoded into 10 features and used as input. Their sensitivity analysis reveals that if different FFNNs are trained for different road types (*e.g.* highways and tunnels), the average performance is better than using a single FFNN for the entire network with road type as one of the inputs.

In Yuan et al. (2018), the authors present Hetero-ConvLSTM, which, as the name suggests, is an extension of the ConvLSTM (Xingjian et al., 2015) architecture. The authors acknowledge the heterogeneous characteristics of the road network and hence, leverage it to produce better performances compared to previous work. They train different neural networks for different parts of the city and then create an ensemble to predict city-wide accident risk. They report an MSE of 0.021 for urban roads and 0.006 for rural roads. For comparison, the LSTM model achieves 0.187 and 0.042 for rural and urban roads respectively. Their dataset consists of 8192 grids with each having an area of 5×5 = 25 square kilometres. It comprises 8 years of data from various sources (weather, traffic, accidents, satellite) in Iowa, USA. The issue of spatial heterogeneity has been taken into account by other research. Some examples are Huang et al. (2019), who propose a hierarchical fusion network to handle the temporal, and spatial heterogeneity of data and Zhou et al. (2019), who use an attention model to preferentially weight the network similarity factors to predict hourly crash risk. Gupta et al. (2019) use an attention model to handle the spatial heterogeneity and report an accuracy of 86.21% using traffic data from London.

Online learning: Fan et al. (2019) propose an *online* FC Deep learning model to predict accident blackspots. They report accuracies for three different types of weather, i.e. 88.70%, 85.19% and 83.33% for sunny, cloudy and windy days respectively.

Table 4
Summary of papers discussed in Section 6.2 (Deep learning for non-recurring congestion alleviation).

Paper	DNN architecture	Predicted variable	Performance	Data source	Unique aspect
(Najjar et al., 2017)	AlexNet, 3 levels	Multi-class classification	73.1%	Satellite imagery	Transfer learning trained on New York validated on Denver
(Gupta et al., 2019)	ResNet	Multi-class classification	74.77% for NY 76.2% for Denver	Satellite imagery	Transfer learning trained on London validated on NY and Denver
Zhao et al. (2018)	FFNN (2 hidden layers)	Crash risk prediction	–	1 year car accidents data from UK	Used PCA to handle correlated readings from nearby sensors
(García de Soto et al., 2018)	FFNN (one hidden layer)	Accident risk for different severity levels	MAPE (based on accident type) Light: 22.40% Severe: 27.00% Fatal: 30.00%	2 years 13109 road segments from Switzerland	Demonstrate that an ensemble of models each trained for a single road type outperforms global model
(Yuan et al., 2018)	(Hetero) ConvLSTM	Accident risk	MSE = 0.021 (urban) MSE = 0.006 (rural)	8192 grids (each 25 sq. km.) Iowa, USA	1. Leveraged heterogeneity of road networks to propose an ensemble 2. Insights into differences between the complexity of urban vs rural
(Fan et al., 2019)	FFNN	Accident black spots	Accuracy (based on weather type) Sunny: 88.70% Cloudy: 85.19% Windy: 83.33%	6391 accident records over 18 months from Jiangsu Province China	1. Online learning 2. Highlights importance of weather conditions

6.3. Potential link between non-recurring and recurring congestion alleviation

The statement '*accidents cause congestion*' is intuitive and undisputed but the reverse causality has not been established in a universal fashion. Retallack and Ostendorf (2019) present a detailed summary of the current understanding of the relationship between traffic congestion and accidents. They report that researchers are far from reaching a consensus on the question, with conflicting reports from studies based on different types of roads and different countries of study. However, they find that a U-shaped curve between congestion levels and accidents appears to be a common observation in recent work which used a sufficiently large number of data points (Fig. 9). A U-shaped curve implies that very high and very low levels of traffic density result in a higher number of accidents. If the U-shaped curve can be reproduced in future research, this can have far-reaching consequences in the attempts to reduce accidents and congestion. As shown in Fig. 9, it would imply that the efforts to reduce accidents would also reduce congestion to levels of maximum flow of traffic (Pasidis, 2019). In addition to the number of accidents, another important factor to consider is the severity of accidents. Establishing the causality behind the severity of accidents is a topic covered in road safety studies and is outside the scope of this survey. The interested reader is referred to (Wang et al., 2013).

7. Challenges and future directions

7.1. Challenges

In this section, we summarise the challenges which were apparent while compiling the research papers for this survey. These challenges have been discussed in detail in the summary of specific sections.

7.1.1. Identifying the state-of-the-art model

In order to establish the state-of-the-art, the relevant work should be given a fair comparison. Fairness can be ensured by comparing the performance using just one parameter at a time — while keeping all the other parameters unchanged. Ensuring such fairness is a challenging task in any field of research. Here, we discuss the challenges specific to the prediction tasks discussed in this survey.

1. *Different definitions of congestion based on different traffic variables:* As discussed in Section 2.3, congestion can be defined using different traffic variables. Due to the lack of a unified relationship between these variables, such differences in the choice of variables make a comparison between the reported prediction performances difficult.

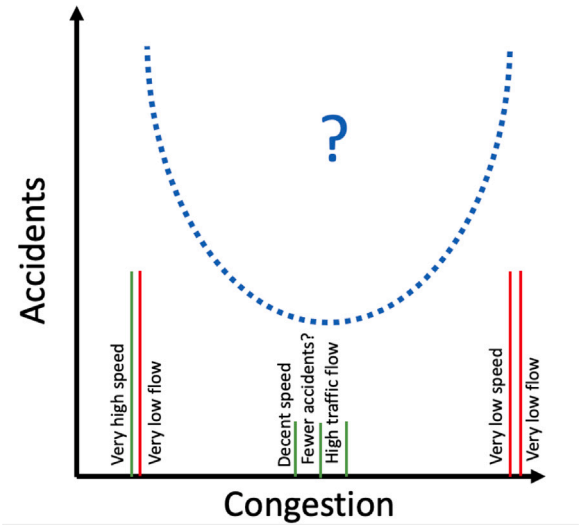

Fig. 9. A hypothetical U-shaped curve between accidents and congestion. Such curves have been observed in recent research.

2. *Different quantisations of the same traffic variable:* Even if two research papers use the same traffic variable to define congestion, their quantisations might differ. For example, if the congestion is defined based on speed, one paper might present a binary prediction task ('jam' or 'no jam'), while the other might present a multi-level variable ('low traffic', 'heavy traffic', 'total jam'). If the dataset is fixed, a binary prediction task is *usually* easier to solve than a multi-class prediction task (Allwein et al., 2000).
3. *Different resolutions of data across countries:* Researchers from different countries might work on the same prediction task but it may be unfair to compare their performance levels due to the differences in data resolution. Such differences in resolution can be attributed to the infrastructure needed for data collection. Data collected from static sensors on roads in developed countries might have a higher resolution in space as well as time. The difference in the resolution of data implies a difference in the complexity of the prediction task at hand. Whether an increase in the resolution of data makes a prediction task easier or more difficult, is an interesting question. Such differences in the relative difficulty of prediction tasks can be better explained by an example. Let us consider two researchers A and B attempting to predict the traffic speed for the next 5 min. A has access to historical traffic speed data at a resolution of 5 s while B has access to historical traffic speed data at a resolution of 1 s. Now, if A and B both attempt to predict the traffic speed at 5-second resolution, B's task is easier compared to A — due to the fact that B has more information for the same task. However, if both A and B attempt to predict the traffic speed at the same resolution as their input data (A: 5 s; B: 1 s), it becomes challenging to comment on whose task is easier. It should be noted that some data sources significantly reduce this problem of differences in the resolution of data. Data collected using distributed sensors (*e.g.* GPS data) have a similar resolution across different countries. An empirical study on such differences in relative difficulties of prediction tasks has been presented in Ma et al. (2015).
4. *Different lengths of prediction horizons:* Varying lengths of the prediction horizon result in different levels of complexity of the prediction task. As noted in Yu et al. (2017), the performance of most models degrades when the length of the prediction horizon increases. This degradation of performance with the increase of prediction horizon is a common finding across research (Wang et al., 2016; Di et al., 2019). We like to highlight the obvious fact that the length of the prediction horizon and usefulness of the prediction task are inversely related.
5. *Different metrics used to measure performance:* The choice of different metrics for performance poses difficulty while comparing different research. The demonstration of superior model performance using one metric does not necessarily imply superior performance when using some other metric. Congestion prediction tasks are examples of imbalanced datasets, thus the most common metrics for classification tasks (for example, accuracy) are usually misleading. In such a scenario, accuracy carries significance only when it is reported along with other metrics carrying complementary information (for example, False Positive rates). A systematic study of the choice of metrics for classification tasks is presented in Sokolova and Lapalme (2009).
6. *Differences in the heterogeneity of traffic stream*: Heterogeneous traffic stream is a common occurrence in developing countries due to a mix of different types of vehicles and high variation in driving behaviour. Correspondingly, we observe a systematic drop in the model performance when deep learning models are used for congestion-related tasks in developing countries. We observed this trend to be valid in each of the congestion detection, prediction and alleviation sections.

A recent attempt to define and maintain a *state-of-the-art* in various domains is https://paperswithcode.com/sota. Their website presents leader boards for different tasks across various domains, including the domain of traffic prediction. This portal is slowly gaining popularity amongst the machine learning community.

7.1.2. Lack of online learning

Online learning (often called continuous learning) refers to the process of updating a model as new data arrives. In the absence of online learning, researchers gather all data before attempting to model the behaviour of the system. They then split the data points into three parts — *train*, *validation*, and *test*. The model is trained using the train data. In order to tune the hyper-parameters (*e.g.* number of layers in the neural network), the model performance is evaluated on the validation data. Finally, after the entire training process is complete, the model performance is evaluated on the test data. The model is said to generalise well if the model performance is similar for both the validation and the test data. The approach mentioned above appears to conform to the standard practices of machine learning. However, an important caveat is that in principle, the labels for the test data should be unseen not only to the *model* but also the *modeller*. When the test data are available to the modeller beforehand, extreme care needs to be taken to ensure that any form of target leakage does not take place. Wujek et al. (2016) summarise the best practices to avoid any unintentional target leakage.

Online learning eliminates this challenge completely. Additionally, online learning helps us model the real world scenario more closely. In the real-world deployment of a traffic prediction model, the test data are available only *after* the deployment of the solution. However, we could not find many examples of research papers using models with online learning capabilities. This might be due to the security and privacy concerns with real-time traffic data being relayed to researchers.

7.2. Conclusions and future research directions

This subsection highlights possible directions for future research. These directions are proposed in answer to the challenges which were presented in the previous subsection. Here, we highlight the importance of data standardisation, potential synergies with other domains of work and the potential synergies between the simulation-based and deep learning approaches for traffic prediction tasks.

7.2.1. Standardised datasets and competitions

The success of deep learning is an example of how good quality data can transform a domain of work—the most notable example is Computer Vision (CV). CV refers to the umbrella term for tasks pertaining to the extraction of information from images. Some examples of specific CV tasks are object localisation, image classification and facial recognition. Given some constraints, the deep learning methods have achieved near-human performance levels for CV tasks. The two biggest impetuses to CV research were the release of the ImageNet (Deng et al., 2009) dataset and its use in the Pascal Visual Object recognition (Everingham et al., 2010). It was noticed that the deep neural networks which had been around for a few decades, started performing very well when trained on ImageNet.

For the domain of traffic congestion prediction, several new possibilities arise with access to standardised datasets. First, the performance of deep learning models improves with increasing amounts of good-quality data. Second, access to common databases can allow the practitioners to establish the state-of-the-art in a disambiguous fashion. Third, it can open avenues for transfer learning where the deep neural network trained on a huge traffic dataset can be readily applied to a new city for which traffic data might not be available (Tan et al., 2018). Apart from these, standardisation also addresses the problem of train–validation–test split as mentioned in the previous subsection, since the test data are not visible to the individuals participating in the competition.

Some exclusive efforts for data standardisation in traffic prediction are (Snyder and Do, 2019), (Wang et al., 2018; Cheng et al., 2018; Moosavi et al., 2019). A list of the latest sources of publicly available traffic data is maintained at the website of 'Research in Intelligent Transportation and Road Safety at Polytechnique Montréal', though videos and images form the majority of the list. Traffic prediction tasks are also gaining popularity in the competition tracks at machine learning conferences. Some recent examples are traffic4cast competition at NIPS, 2020 and Night-owls competition at CVPR, 2020.

7.2.2. Similar prediction tasks in other domains

Papers using machine learning for internet traffic classification have been popular in the Wireless Networks community for a long time. Some prediction tasks in wireless networks are similar to traffic prediction. Survey papers from the wireless network domain might hold a treasure trove of ideas for transportation researchers. Some examples of relevant surveys include (Mao et al., 2018; Zhang et al., 2019b; Nguyen and Armitage, 2008). A routing protocol is the counterpart of a traffic signal control strategy in transportation networks. Hence, protocol-free wireless routing is akin to adaptive signal control. Making wireless networks free from (fixed) protocols is discussed in Tang et al. (2017). Yu et al. (2019) investigate using deep-reinforcement learning to handle the heterogeneity in wireless networks. Similarly, congestion-aware routing of wireless networks (Chen et al., 2007) is the counterpart of congestion-alleviation algorithms in transportation. Weather-forecasting is another domain where the prediction tasks and datasets are similar to transportation. A classic example is (Xingjian et al., 2015), which has resulted in a large number of applications in traffic (Wang et al., 2020a).

7.2.3. Synergies between model-driven and deep learning based approaches

Deep neural networks have high predictive power and can be trained much faster and with little effort compared to calibrating a traffic simulator. Deep learning models lack interpretability. The inherent interpretable nature of traffic simulators can be leveraged by deep learning researchers when both methods are used in conjunction. Such synergies are already being explored in deep reinforcement learning-based approaches for traffic signal control. Some researchers have used microscopic traffic simulators to generate data for congestion labels, thereby solving the class imbalance problem. More such efforts in the direction of creating synergies between the two approaches are needed.

Acknowledgements

This work is an outcome of the Future Resilient Systems project at the Singapore-ETH Centre (SEC) supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. The authors would like to acknowledge the valuable feedback and recommendations received from Dr. Yi Wang from Future Resilient Systems and Dr. Jimi B. Oke from University of Massachusetts, Amherst. The authors would also like to acknowledge the three anonymous reviewers for their valuable comments and suggestions.

References

Adnan, M., Pereira, F.C., Azevedo, C.M.L., Basak, K., Lovric, M., Raveau, S., Zhu, Y., Ferreira, J., Zegras, C., Ben-Akiva, M., 2016. Simmobility: A multi-scale integrated agent-based simulation platform. In: 95th Annual Meeting of the Transportation Research Board.

Afrin, T., Yodo, N., 2020. A survey of road traffic congestion measures towards a sustainable and resilient transportation system. Sustainability 12 (11), 4660.

Albawi, S., Mohammed, T.A., Al-Zawi, S., 2017. Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology. ICET, IEEE, pp. 1–6.

Aljamal, M.A., Rakha, H.A., Du, J., El-Shawarby, I., 2018. Comparison of microscopic and mesoscopic traffic modeling tools for evacuation analysis. In: 2018 21st International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 2321–2326.

Allwein, E.L., Schapire, R.E., Singer, Y., 2000. Reducing multiclass to binary: A unifying approach for margin classifiers. J. Mach. Learn. Res. 1 (Dec), 113–141.

Andreoletti, D., Troia, S., Musumeci, F., Giordano, S., Maier, G., Tornatore, M., 2019. Network traffic prediction based on diffusion convolutional recurrent neural networks. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops. INFOCOM WKSHPS, IEEE, pp. 246–251.

Axhausen, K., Horni, A., Nagel, K., 2016. The Multi-Agent Transport Simulation MATSim. Ubiquity Press.

Balakrishna, R., Ben-Akiva, M., Bottom, J., Gao, S., 2013. Information impacts on traveler behavior and network performance: State of knowledge and future directions. Adv. Dyn. Netw. Model. Complex Transp. Syst. 193–224.

Barth, M., Boriboonsomsin, K., 2009. Traffic congestion and greenhouse gases. Access Mag. 1 (35), 2–9.

Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D., 2011. SUMO–simulation of urban mobility: an overview. In: Proceedings of SIMUL 2011, the Third International Conference on Advances in System Simulation. ThinkMind.

Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H., Mishalani, R., 1998. DynaMIT: a simulation-based system for traffic prediction. In: DACCORD Short Term Forecasting Workshop. Citeseer, pp. 1–12.

Ben-David, A., 2008. Comparison of classification accuracy using Cohen's weighted kappa. Expert Syst. Appl. 34 (2), 825–832.

Borysov, S.S., Rich, J., Pereira, F.C., 2019. How to generate micro-agents? A deep generative modeling approach to population synthesis. Transp. Res. C 106, 73–97.

Casas, J., Ferrer, J.L., Garcia, D., Perarnau, J., Torday, A., 2010. Traffic simulation with aimsun. In: Fundamentals of Traffic Simulation. Springer, pp. 173–232.

Cassidy, M.J., Bertini, R.L., 1999. Some traffic features at freeway bottlenecks. Transp. Res. B 33 (1), 25–42.

Chakraborty, P., Adu-Gyamfi, Y.O., Poddar, S., Ahsani, V., Sharma, A., Sarkar, S., 2018. Traffic congestion detection from camera images using deep convolution neural networks. Transp. Res. Rec. 2672 (45), 222–231.

Chen, X., Jones, H., Jayalath, A., 2007. Congestion-aware routing protocol for mobile ad hoc networks. In: 2007 IEEE 66th Vehicular Technology Conference, pp. 21–25.

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

Chung, Y., 2011. Quantification of nonrecurrent congestion delay caused by freeway accidents and analysis of causal factors. Transp. Res. Rec. 2229 (1), 8–18.

Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Connie, L., On, C.K., Patricia, A., 2018. A review of automatic license plate recognition system in mobile based platform. J. Telecommun. Electron. Comput. Eng. 10 (3–2), 77–82.

Daganzo, C., Daganzo, C., 1997. Fundamentals of Transportation and Traffic Operations, Vol. 30. Pergamon Oxford.

De Fabritiis, C., Ragona, R., Valenti, G., 2008. Traffic estimation and prediction based on real time floating car data. In: 2008 11th International IEEE Conference on Intelligent Transportation Systems. IEEE, pp. 197–203.

de Palma, A., Lindsey, R., 2011. Traffic congestion pricing methodologies and technologies. Transp. Res. C 19 (6), 1377–1399.

de Palma, A., Proost, S., Seshadri, R., Ben-Akiva, M., 2018. Congestion tolling-dollars versus tokens: A comparative analysis. Transp. Res. B 108, 261–280.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 248–255.

Di, X., Xiao, Y., Zhu, C., Deng, Y., Zhao, Q., Rao, W., 2019. Traffic congestion prediction by spatiotemporal propagation patterns. In: 2019 20th IEEE International Conference on Mobile Data Management. MDM, IEEE, pp. 298–303.

Ding, W., Jing, X., Yan, Z., Yang, L.T., 2019. A survey on data fusion in internet of things: Towards secure and privacy-preserving fusion. Inf. Fusion 51, 129–144.

Dusparic, I., Monteil, J., Cahill, V., 2016. Towards autonomic urban traffic control with collaborative multi-policy reinforcement learning. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems, ITSC, pp. 2065–2070.

Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. 88 (2), 303–338.

Fan, Z., Liu, C., Cai, D., Yue, S., 2019. Research on black spot identification of safety in urban traffic accidents based on machine learning method. Saf. Sci. 118, 607–616.

Fouladgar, M., Parchami, M., Elmasri, R., Ghaderi, A., 2017. Scalable deep traffic flow neural networks for urban traffic congestion prediction. In: 2017 International Joint Conference on Neural Networks, IJCNN, pp. 2251–2258.

Fukuda, S., Uchida, H., Fujii, H., Yamada, T., 2020. Short-term prediction of traffic flow under incident conditions using graph convolutional recurrent neural network and traffic simulation. IET Intell. Transp. Syst.

Gao, J., Li, P., Chen, Z., Zhang, J., 2020. A survey on deep learning for multimodal data fusion. Neural Comput. 32 (5), 829–864.

Gartner, N.H., Messer, C.J., Rathi, A., 2002. Traffic flow theory-a state-of-the-art report: revised monograph on traffic flow theory. In: Transportation Research Board (TRB) Special Report. Turner-Fairbank Highway Research Center.

Genders, W., Razavi, S., 2016. Using a deep reinforcement learning agent for traffic signal control. arXiv preprint arXiv:1611.01142.

Genders, W., Razavi, S., 2019. Asynchronous n-step q-learning adaptive traffic signal control. J. Intell. Transp. Syst. 23 (4), 319–331.

Gerlough, D.L., Huber, M.J., 1976. Traffic flow theory. In: Transportation Research Board (TRB) Special Report. Transportation Research Board.

Geroliminis, N., Sun, J., 2011. Hysteresis phenomena of a macroscopic fundamental diagram in freeway networks. Procedia Soc. Behav. Sci. 17, 213–228.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587.

Golledge, R.G., 1995. Path selection and route preference in human navigation: A progress report. In: International Conference on Spatial Information Theory. Springer, pp. 207–222.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.
Google, 2020. How AI helps predict traffic and determine routes. Internet.
Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J., 2016. LSTM: A search space odyssey. IEEE Trans. Neural Netw. Learn. Syst. 28 (10), 2222–2232.
Gregurić, M., Kušić, K., Vrbanić, F., Ivanjko, E., 2020. Variable speed limit control based on deep reinforcement learning: A possible implementation. In: 2020 International Symposium ELMAR. pp. 67–72.
Gupta, S., Srivatsav, D., Subramanyam, A., Kumaraguru, P., 2019. Attentional road safety networks. In: 2019 IEEE International Conference on Image Processing ICIP, pp. 1600–1604.
Haberman, S.J., 2019. Measures of agreement versus measures of prediction accuracy. ETS Res. Rep. Ser. 2019 (1), 1–23.
Hall, F.L., 1996. Traffic stream characteristics. In: Traffic Flow Theory, Vol. 36. US Federal Highway Administration.
Hallenbeck, M.E., Ishimaru, J., Nee, J., et al., 2003. Measurement of recurring versus non-recurring congestion. Technical Report, Washington (State). Dept. of Transportation.
He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.
Hennessy, D.A., Wiesenthal, D.L., 1999. Traffic congestion, driver stress, and driver aggression. Aggress. Behav. Official J. Int. Soc. Res. Aggress. 25 (6), 409–423.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.
Huang, J., Agarwal, S., 2020. Physics informed deep learning for traffic state estimation. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 1–6.
Huang, C., Zhang, C., Dai, P., Bo, L., 2019. Deep dynamic fusion network for traffic accident forecasting. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 2673–2681.
Immers, L., Logghe, S., 2002. Traffic Flow Theory, Vol. 40 (21). Faculty of Engineering, Department of Civil Engineering, Section Traffic and Infrastructure, Kasteelpark Arenberg.
Impedovo, D., Balducci, F., Dentamaro, V., Pirlo, G., 2019. Vehicular traffic congestion classification by visual features and deep learning approaches: a comparison. Sensors 19 (23), 5213.
Janson, B.N., 1991. Dynamic traffic assignment for urban road networks. Transp. Res. B 25 (2–3), 143–161.
Jiang, H., Sun, D., Jampani, V., Yang, M.-H., Learned-Miller, E., Kautz, J., 2018. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9000–9008.
Jozefowicz, R., Zaremba, W., Sutskever, I., 2015. An empirical exploration of recurrent network architectures. In: International Conference on Machine Learning. PMLR, pp. 2342–2350.
Khan, S.D., Ullah, H., 2019. A survey of advances in vision-based vehicle re-identification. Comput. Vis. Image Underst. 182, 50–63.
Kleinrock, L., 2018. Internet congestion control using the power metric: Keep the pipe just full, but no fuller. Ad Hoc Netw. 80, 142–157.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 1097–1105.
Kurniawan, J., Syahra, S.G., Dewa, C.K., Afiahayati, 2018. Traffic congestion detection: Learning from CCTV monitoring images using convolutional neural network. Procedia Comput. Sci. 144, 291–297, INNS Conference on Big Data and Deep Learning.
Kyaw, N.N., Sinha, G., Mon, K.L., 2018. License plate recognition of myanmar vehicle number plates a critical review. In: 2018 IEEE 7th Global Conference on Consumer Electronics. GCCE, IEEE, pp. 771–774.
LeCun, Y., Touresky, D., Hinton, G., Sejnowski, T., 1988. A theoretical framework for back-propagation. In: Proceedings of the 1988 Connectionist Models Summer School, vol. 1, pp. 21–28.
LeNail, A., 2019. Nn-svg: Publication-ready neural network architecture schematics. J. Open Source Softw. 4 (33), 747.
Li, Y., Guo, Y., Lu, J., Peeta, S., 2019. Impacts of congestion pricing and reward strategies on automobile travelers' morning commute mode shift decisions. Transp. Res. A 125, 72–88.
Li, G., Lai, W., Sui, X., Li, X., Qu, X., Zhang, T., Li, Y., 2020a. Influence of traffic congestion on driver behavior in post-congestion driving. Accid. Anal. Prev. 141, 105508.
Li, L., Sheng, X., Du, B., Wang, Y., Ran, B., 2020b. A deep fusion model based on restricted Boltzmann machines for traffic accident duration prediction. Eng. Appl. Artif. Intell. 93, 103686.
Li, Y., Yu, R., Shahabi, C., Liu, Y., 2017a. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926.
Lin, Y., Li, R., 2020. Real-time traffic accidents post-impact prediction: Based on crowdsourcing data. Accid. Anal. Prev. 145, 105696.
Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context. In: European Conference on Computer Vision. Springer, pp. 740–755.
Litman, T., 2016. Smart congestion relief: Comprehensive analysis of traffic congestion costs and congestion reduction benefits.
Liu, J., Barreau, M., Cicic, M., Johansson, K.H., 2020. Learning-based traffic state reconstruction using probe vehicles. arXiv preprint arXiv:2011.05031.
Liu, Y., Zheng, H., Feng, X., Chen, Z., 2017. Short-term traffic flow prediction with conv-LSTM. In: 2017 9th International Conference on Wireless Communications and Signal Processing. WCSP, IEEE, pp. 1–6.
Ma, X., Yu, H., Wang, Y., Wang, Y., 2015. Large-scale transportation network congestion evolution prediction using deep learning theory. PLoS One 10 (3), e0119044.
Mahmassani, H.S., Sbayti, H., Zhou, X., 2004. DYNASMART-P Version 1.0 User's Guide, Vol. 137. Maryland Transportation Initiative, College Park, Maryland.
Manifest, 2018. The popularity of google maps: Trends in navigation apps in 2018. Internet.
Mao, Q., Hu, F., Hao, Q., 2018. Deep learning for intelligent wireless networks: A comprehensive survey. IEEE Commun. Surv. Tutor. 20 (4), 2595–2621.
McGroarty, J., 2010. Recurring and Non-Recurring Congestion: causes, Impacts, and Solutions. University of Cincinnati, Neihoff Urban Studio–W10.
Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533.
Mo, Z., Di, X., Shi, R., 2020. A physics-informed deep learning paradigm for car-following models. arXiv preprint arXiv:2012.13376.
Mohamed, I.S., 2017. Detection and Tracking of Pallets Using a Laser Rangefinder and Machine Learning Techniques (Ph.D. thesis). European Master on Advanced Robotics+(EMARO+), University of Genova, Italy.
Moosavi, S., Samavatian, M.H., Parthasarathy, S., Teodorescu, R., Ramnath, R., 2019. Accident risk prediction based on heterogeneous sparse data: New dataset and insights. In: Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 33–42.
Nagel, K., Rickert, M., 2001. Parallel implementation of the TRANSIMS micro-simulation. Parallel Comput. 27 (12), 1611–1639.
Najjar, A., Kaneko, S., Miyanaga, Y., 2017. Combining satellite imagery and open data to map road safety. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pp. 4524–4530.
Newell, G.F., 1989. Theory of highway traffic signals. ITS Rep. 1989 (07).
Nguyen, T., Armitage, G., 2008. A survey of techniques for internet traffic classification using machine learning. IEEE Commun. Surv. Tutor. 10 (4), 56–76.
Nguyen, H., Kieu, L.-M., Wen, T., Cai, C., 2018. Deep learning methods in transportation domain: a review. IET Intell. Transp. Syst. 12 (9), 998–1004.
Pasidis, I., 2019. Congestion by accident? A two-way relationship for highways in England. J. Transp. Geogr. 76, 301–314.

Polson, N.G., Sokolov, V.O., 2017. Deep learning for short-term traffic flow prediction. Transp. Res. C 79, 1–17.
Rahman, R., Hasan, S., 2020. Real-time signal queue length prediction using long short-term memory neural network. Neural Comput. Appl. 1–14.
Raissi, M., Perdikaris, P., Karniadakis, G.E., 2017a. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711.10561.
Raissi, M., Perdikaris, P., Karniadakis, G.E., 2017b. Physics informed deep learning (Part II): Data-driven discovery of nonlinear partial differential equations.
Ramezani, M., Haddad, J., Geroliminis, N., 2015. Dynamics of heterogeneity in urban networks: aggregated traffic modeling and hierarchical control. Transp. Res. B 74, 1–19.
Ranjan, N., Bhandari, S., Zhao, H.P., Kim, H., Khan, P., 2020. City-wide traffic congestion prediction based on CNN, LSTM and transpose CNN. IEEE Access 8, 81606–81620.
Rashmi, C., Shantala, C., 2020. Vehicle density analysis and classification using YOLOv3 for smart cities. In: 2020 4th International Conference on Electronics, Communication and Aerospace Technology. ICECA, IEEE, pp. 980–986.
Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788.
Ren, H., Song, Y., Wang, J., Hu, Y., Lei, J., 2018. A deep learning approach to the citywide traffic accident risk prediction. In: 2018 21st International Conference on Intelligent Transportation Systems, ITSC, pp. 3346–3351.
Retallack, A.E., Ostendorf, B., 2019. Current understanding of the effects of congestion on traffic accidents. Int. J. Environ. Res. Public Health 16 (18), 3400.
Rinne, H., 2008. The Weibull Distribution: A Handbook. CRC Press.
Rummery, G.A., Niranjan, M., 1994. On-Line Q-Learning using Connectionist Systems, Vol. 37. Citeseer.
Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Netw. 61, 85–117.
Schrank, D., Lomax, T., Eisele, B., 2012. 2012 Urban Mobility Report. Texas Transportation Institute, [Online]. Available: http://mobility.tamu.edu/ums/report.
Shabestray, S.M.A., Abdulhai, B., 2019. Multimodal intelligent deep (MiND) traffic signal controller. In: 2019 IEEE Intelligent Transportation Systems Conference. ITSC, IEEE, pp. 4532–4539.
Shi, R., Mo, Z., Di, X., 2021a. Physicsinformed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35 (1), pp. 540–547.
Shi, R., Mo, Z., Huang, K., Di, X., Du, Q., 2021b. Physics-informed deep learning for traffic state estimation. arXiv preprint arXiv:2101.06580.
Shin, D.-H., Chung, K., Park, R.C., 2020. Prediction of traffic congestion based on LSTM through correction of missing temporal and spatial data. IEEE Access 8, 150784–150796.
Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al., 2017. Mastering the game of go without human knowledge. Nature 550 (7676), 354–359.
Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
Smith, L., Beckman, R., Baggerly, K., 1995. TRANSIMS: Transportation Analysis and Simulation System. Technical Report, Los Alamos National Lab., NM (United States).
Snyder, C., Do, M., 2019. STREETS: A novel camera network dataset for traffic flow. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 10242–10253.
Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Inf. Process. Manage. 45 (4), 427–437.
García de Soto, B., Bumbacher, A., Deublein, M., Adey, B.T., 2018. Predicting road traffic accidents using artificial neural network models. Infrastruct. Asset Manag. 5 (4), 132–144.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.
Stopher, P.R., 2004. Reducing road congestion: a reality check. Transp. Policy 11 (2), 117–131.
Sun, S., Chen, J., Sun, J., 2019. Traffic congestion prediction based on GPS trajectory data. Int. J. Distrib. Sens. Netw. 15 (5), 1550147719847440.
Sun, F., Dubey, A., White, J., 2017. Dxnat—deep neural networks for explaining non-recurring traffic congestion. In: 2017 IEEE International Conference on Big Data. Big Data, IEEE, pp. 2141–2150.
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.
Szepesvári, C., 2010. Algorithms for reinforcement learning. Synth. Lect. Artif. Intell. Mach. Learn. 4 (1), Abstract section.
Tadikamalla, P.R., 1980. A look at the burr and related distributions. Int. Stat. Rev./Rev. Int. Stat. 337–344.
Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C., 2018. A survey on deep transfer learning. In: International Conference on Artificial Neural Networks. Springer, pp. 270–279.
Tang, F., Mao, B., Fadlullah, Z.M., Kato, N., Akashi, O., Inoue, T., Mizutani, K., 2017. On removing routing protocol from future wireless networks: A real-time deep learning approach for intelligent traffic control. IEEE Wirel. Commun. 25 (1), 154–160.
Tian, Y., Zhang, K., Li, J., Lin, X., Yang, B., 2018. LSTM-based traffic flow prediction with missing data. Neurocomputing 318, 297–305.
Tillema, T., Ben-Elia, E., Ettema, D., van Delden, J., 2013. Charging versus rewarding: A comparison of road-pricing and rewarding peak avoidance in the Netherlands. Transp. Policy 26, 4–14.
Treiterer, J., 1975. Investigation of Traffic Dynamics by Aerial Photogrammetry Techniques. Technical Report.
van Wageningen-Kessels, F., Van Lint, H., Vuik, K., Hoogendoorn, S., 2015. Genealogy of traffic flow models. EURO J. Transp. Logist. 4 (4), 445–473.
Wang, S., Cao, J., Yu, P., 2020a. Deep learning for spatio-temporal data mining: A survey. IEEE Trans. Knowl. Data Eng..
Wang, Z., Chen, J., Hoi, S.C., 2020b. Deep learning for image super-resolution: A survey. IEEE Trans. Pattern Anal. Mach. Intell..
Wang, J., Gu, Q., Wu, J., Liu, G., Xiong, Z., 2016. Traffic speed prediction and congestion source exploration: A deep learning method. In: 2016 IEEE 16th International Conference on Data Mining. ICDM, IEEE, pp. 499–508.
Wang, R., Kashinath, K., Mustafa, M., Albert, A., Yu, R., 2020. Towards physics-informed deep learning for turbulent flow prediction. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1457–1466.
Wang, P., Li, L., Jin, Y., Wang, G., 2018. Detection of unwanted traffic congestion based on existing surveillance system using in freeway via a CNN-architecture trafficnet. In: 2018 13th IEEE Conference on Industrial Electronics and Applications, ICIEA, pp. 1134–1139.
Wang, C., Quddus, M.A., Ison, S.G., 2013. The effect of traffic and road characteristics on road safety: A review and future research direction. Saf. Sci. 57, 264–275.
Wang, Q., Wan, J., Yuan, Y., 2018. Locality constraint distance metric learning for traffic congestion detection. Pattern Recognit. 75, 272–281.
Wang, Y., Zhang, D., Liu, Y., Dai, B., Lee, L.H., 2019. Enhancing transportation systems via deep learning: A survey. Transp. Res. C 99, 144–163.
Watkins, C.J.C.H., 1989. Learning from Delayed Rewards. King's College, Cambridge.
Weisbrod, G., Vary, D., Treyz, G., 2001. Economic Implications of Congestion. Project A2-21 FY'97.
Wu, Y.-W., Hsu, T.-P., 2021. Mid-term prediction of at-fault crash driver frequency using fusion deep learning with city-level traffic violation data. Accid. Anal. Prev. 150, 105910.
Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S., 2020. A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. 1–21.
Wu, Y., Tan, H., Qin, L., Ran, B., 2020a. Differential variable speed limits control for freeway recurrent bottlenecks via deep actor-critic algorithm. Transp. Res. C 117, 102649.

Wu, T., Zhou, P., Liu, K., Yuan, Y., Wang, X., Huang, H., Wu, D.O., 2020b. Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks. IEEE Trans. Veh. Technol..

Wujek, B., Hall, P., Güneş, F., 2016. Best Practices for Machine Learning Applications. SAS Institute Inc.

Xie, P., Li, T., Liu, J., Du, S., Yang, X., Zhang, J., 2020. Urban flow prediction from spatiotemporal data using machine learning: A survey. Inf. Fusion 59, 1–12.

Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., Woo, W.-c., 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems. pp. 802–810.

Yen, C.-C., Ghosal, D., Zhang, M., Chuah, C.-N., 2020. A deep on-policy learning agent for traffic signal control of multiple intersections. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 1–6.

Yoshimura, S., 2006. MATES: Multi-agent based traffic and environment simulator-theory, implementation and practical application. Comput. Model. Eng. Sci. 11 (1), 17–25.

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

Yu, Y., Wang, T., Liew, S.C., 2019. Deep-reinforcement learning multiple access for heterogeneous wireless networks. IEEE J. Sel. Areas Commun. 37 (6), 1277–1290.

Yuan, Z., Zhou, X., Yang, T., 2018. Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 984–992.

Zeng, J., Hu, J., Zhang, Y., 2018. Adaptive traffic signal control with deep recurrent Q-learning. In: 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, pp. 1215–1220.

Zhang, J., Junhua, W., Shou'en, F., 2019a. Prediction of urban expressway total traffic accident duration based on multiple linear regression and artificial neural network. In: 2019 5th International Conference on Transportation Information and Safety. ICTIS, IEEE, pp. 503–510.

Zhang, C., Patras, P., Haddadi, H., 2019b. Deep learning in mobile and wireless networking: A survey. IEEE Commun. Surv. Tutor. 21 (3), 2224–2287.

Zhao, H., Mao, T., Yu, H., Zhang, M.K., Zhu, H., 2018. A driving risk prediction algorithm based on PCA-BP neural network in vehicular communication. In: 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics. Vol. 2. IHMSC, IEEE, pp. 164–169.

Zhou, Z., Chen, L., Zhu, C., Wang, P., 2019. Stack ResNet for short-term accident risk prediction leveraging cross-domain data. In: 2019 Chinese Automation Congress. CAC, IEEE, pp. 782–787.