Transportation Research Part C 124 (2021) 102977

Contents lists available at ScienceDirect

Transportation Research Part C

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

# Transferability improvement in short-term traffic prediction using stacked LSTM network

Junyi Li [a,b,c,*], Fangce Guo [b], Aruna Sivakumar [b], Yanjie Dong [b], Rajesh Krishnan [b]

[a] College of Civil Engineering and Architecture, Zhejiang University, Hangzhou, China
[b] Centre for Transport Studies, Imperial College London, South Kensington Campus, London SW7 2AZ, United Kingdom
[c] Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, Hangzhou, China

ARTICLE INFO

*Keywords:*
Short-term traffic prediction
Transfer learning
Machine learning methods
Model transferability
Stacked LSTM network

ABSTRACT

Short-term traffic flow forecasting is a key element in Intelligent Transport Systems (ITS) to provide proactive traffic state information to road network operators. A variety of methods to predict traffic variables in the short-term can be found in the literature, ranging from time-series algorithms, machine learning tools and deep learning methods to a selective hybrid of these approaches. Despite the advances in prediction techniques, a challenging problem that affects the application of such methods in the real world is the prevalence of insufficient data across an entire network. It is rare that extensive historical training data required for model training are available for all the links in a city. In order to address this data insufficiency problem, this paper applies transfer learning techniques to machine learning methods in short-term traffic prediction. All the traffic data used in this paper were collected from Highways England road networks in the UK. The results show that through improving the transferability of machine learning-based models, the computational burden due to the model training process can be significantly reduced and the prediction accuracy under data deficient scenarios can be improved for one-step ahead prediction. However, the prediction accuracy gradually decreases in multi-step ahead prediction. It is also found that the accuracy of the proposed hybrid method is highly dependent upon consistency between datasets but less dependent on geographical attributes of links.

## 1. Introduction

Short-term traffic prediction is an essential element in Intelligent Transport Systems (ITS), such as Advanced Traveller Information Systems and Urban Traffic Control (UTC). Specifically, accurate prediction of traffic flow is important for traffic planning and proactive traffic management. Therefore, short-term traffic prediction is a widely researched topic.

In the past decades, many short-term traffic prediction models have been developed based on statistical models and advanced machine learning tools. Given the capability of machine learning methods for handling missing data, as well as their computational efficiency, reliability and robustness, advanced machine learning methods have been widely used in this research area. However, a challenging problem that affects the application of these methods in the real world is the prevalence of insufficient data across an entire network (Vlahogianni et al., 2014). It is rare that extensive historical training data required for model training are available for all the links in a city (Abadi et al., 2014).

* Corresponding author at: B818 Anzhong Building, College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China.
*E-mail address:* junyili@zju.edu.cn (J. Li).

https://doi.org/10.1016/j.trc.2021.102977
Received 15 February 2020; Received in revised form 22 December 2020; Accepted 7 January 2021
Available online 21 January 2021

A major assumption in many machine learning and data mining algorithms is that the training and testing data must be in the same feature space with the same distribution (Pan and Yang, 2009) and extensive historical training data are required for model calibration. In practical applications, this assumption is not easy to be fully meet. Although sufficient data might be available for some highly instrumented arterial roads, it is impossible to obtain data for all minor roads. Additionally, even though sufficient data may be available across the whole urban network, training such a huge model link by link for large urban areas with thousands of distinct links would be extremely time consuming and computationally inefficient. For example, within a 6-mile radius from the centre of London, there are approximately 14,842 km roads distributed over 60,000 minor links with hundreds of different road attributes; however, the network is instrumented only with approximately 6000 inductive loop detectors and hundreds of additional detectors for monitoring and operational purposes. Insufficient data is a practical problem in real-world systems due to a number of reasons, such as newly installed sensors, sensor errors or communication faults, resulting in lack of sufficient historical data for model training. Therefore, it is infeasible both economically and computationally to train such a large number of prediction models using the required historical training data at a link-by-link level on large road networks.

An approach that has been widely used to obtain enough training data is to impute missing data (Laña et al., 2018). Based on different missing patterns, a wide range of researches have been done in the missing data imputation to deal with the data missing problem in the literature, from statistical methods to machine learning tools (Chen et al., 2001; Cui et al., 2019; Laña et al., 2018; Boquet et al., 2020). Most researches on missing data cleaning and imputation in traffic engineering focus on the imputation of missing points and intervals (Little and Rubin, 1987) and spatial–temporal relationships are used to reconstruct incomplete traffic data to handle missing traffic data problems (Van Lint, 2004; Treiber and Helbing, 2002). On one hand, historical observations can be used to replace missing data using heuristic and regression methods, such as historical average, last observation carried forward, regression models, making use of temporal features (Guo et al., 2017b). On the other hand, traffic data collected from nearby locations can be used to impute missing data of the target location, making use of spatio-temporal features. For example, Yin et al. (2002) used upstream flows in the current time interval to forecast the downstream traffic flow. Stathopoulos and Karlaftis (2001) found that considering traffic data from two adjacent sensors would lead to considerable improvement in prediction accuracy. However, both a large amount and high quality of nearby traffic data are required to provide necessary information for the prediction task of target links. El Esawey et al. (2015) found that considering adjacent detectors would not lead to significant accuracy improvement unless the number of adjacent detectors considered exceeds four. In practice, it is not easy to collect real-time high-quality traffic data from more than four detectors near a data-missing link and any data missing conditions in nearby detectors is likely to repeat and lead to a vicious cycle.

Recently, a widely accepted solution toward severe data missing problem is to build up a less-data-demanding model structure (Kisgyörgy and Rilett, 2002; Kumar and Vanajakshi, 2015; Padiath et al., 2009; Badhrudeen et al., 2016). Generally, a preliminary parameter sensitivity analysis should be carried out to select the most appropriate model structure, where the model with the fewest parameters that can adequately describe the process is preferred (Géron, 2019; VanderPlas, 2016). For example, Padiath et al. (2009) chose a 3-layer Neural Network model with 5 neurons in each hidden layer, where the parameters are selected by trial and error for the best result, to address the traffic prediction problem under severe data missing scenarios.

Most of the previous works have focused on data imputation of missing points and intervals. However, a challenging and practical problem of short-term traffic prediction is the prevalence of insufficient training data with continuous missing for a long period in the real-world application. Surprisingly, little attention has been paid in academic studies to traffic prediction in the short-term with insufficient and incomplete data. To the best of authors' knowledge, existing studies on data missing imputation in traffic engineering cannot properly address the problem of insufficient data in consecutive months. In this paper, a short-term traffic prediction framework is proposed to deal with insufficient traffic historical data problems.

The challenge of short-term prediction under the conditions of insufficient data is not unique to transport and it occurs in many other domains, for example, natural language processing (NLP) (see Huang et al. (2013)), energy prediction (see Hu et al. (2016)), recommender systems (see Xin et al. (2014)) and human activity recognition (see Chen et al., 2017a), which also needs complex models and a massive amount of data. For example, in NLP, many works emphasize transferring knowledge between languages due to the small amount target language training data (e.g. Huang et al., 2013; Vu et al., 2014). With such a transfer, a language adaptation model can be established even with very few target training data. In image processing, Yosinski et al. (2014) investigated the transferability of convolutional neural networks and demonstrated that initialising with transferred features can significantly reduce training time and improve generalisation performance. More recently, Segev et al. (2016) developed two transfer learning algorithms based on Random Forests (RF) that utilised a model trained over the source domain and effectively adapted it to the target domain using local adjustments of the tree parameters. In the area of energy, a practical problem in short-term wind speed prediction is insufficient historical data in newly-built wind farms to train prediction models. In order to address this problem, Hu et al. (2016) proposed a transfer learning method with Autoencoder. The proposed model was trained using wind speed data from three data-rich farms with one-year historical records, to extract wind speed patterns, and then finely tune the mapping using data from a newly-built farm with 0.5-month wind speed records. The results showed that prediction errors were significantly reduced using the proposed transfer technique. These transfer learning applications above are quite referential and give us more insights into dealing with the data insufficient problems in traffic prediction.

Although the hybrid method with transfer learning and neural networks to improve model transferability has been demonstrated in other research domains, it is still at an early stage of development in transport (Luan et al., 2018). These above studies show that the approach to improve prediction performance in the case of insufficient data scenarios is to enhance the transferability of machine learning models. The fundamental idea is that since most prediction methods capture underlying traffic abstractions by calibrating inherent parameters, if a shared model from a set of similar links can be trained to generalise intermediate patterns of traffic state

variables that are shared and useful across links, then this predictive shared model should be applicable to other similar links. With such a transfer, the applicability of machine learning methods in practical short-term traffic flow prediction will considerably increase and the prediction on a network scale is more feasible. In this case, knowledge transfer, if applied successfully, could greatly improve the performance of learning under data insufficiency scenarios and accordingly avoid expensive data-labelling efforts, installation of data collection devices and heavy computational burden. Transfer learning methods have been demonstrated with the potential to relax the stringent independent and identical distribution (iid) assumption in conventional machine learning techniques, which can be used to improve the generalisability and speed up the training process. The improved generalisability due to transfer learning methods has the potential to address insufficient training data problems by improving the transferability of a traffic prediction model, and the quicker training process is able to significantly alleviate the related computational burden.

Given this motivation, the main research question in this paper is: whether transfer learning techniques able to address the data insufficient problem in traffic prediction and what are the significant factors that affect the accuracy of short-term prediction models using hybrid methods combining transfer learning with machine learning? In order to answer the questions, the specific objectives of this paper are 1) to investigate the transferability of machine learning methods in short-term traffic prediction using real-world datasets with insufficient training data; 2) to use transfer learning techniques with deep learning tools to improve the transferability of machine learning models in short-term traffic prediction, thus enhancing the model accuracy and reducing the model training time; 3) to evaluate the impacts of different data selection criteria in the transferring process and 4) to investigate the relationship between the accuracy of the proposed model and the prediction horizon.

## 2. Previous studies

### *2.1. Short-term traffic prediction with machine learning tools*

A large number of machine learning-based models in short-term traffic prediction have been published over the last four decades, such as Neural Networks (NN) (Yasdi, 1999; Innamaa, 2000; Ishak et al., 2003; Zhang and Zhang, 2016; Pamuła, 2019; Zhou et al., 2020), Support Vector Regression (SVR) (Lippi et al., 2013; Guo et al., 2018), Random Forests and k-Nearest Neighbours (Habtemichael and Cetin, 2016; Guo et al., 2017). The main advantages of machine learning-based prediction models are that these models are less complicated to implement in different contents and have more accurate prediction results.

Recently, Deep Neural Networks (DNNs) become widely used for both link-level (e.g. Ma et al. (2015); Huang et al. (2014)) and network-level prediction (e.g. Laharotte et al. (2015); Vaughan et al. (2013); Cui et al. (2018); Zhang et al. (2019); Wang et al. (2019)) in the short-term in traffic prediction field due to its capacity for modelling complex non-linear relationships, especially spatio-temporal relationships within historical traffic profiles, to increase the prediction accuracy of models. Compared with conventional shallow neural networks, the extra layers of DNNs enable more feedback loops and composition of features from lower layers, leading to better learning capacity and generalization capacity (Cui et al., 2018; Vinayakumar et al., 2017; Du et al., 2017; Huang et al., 2014). For example, Huang et al. (2014) proposed a Deep Belief Network (DBN) and Multitask Learning (MTL) method in traffic flow prediction, which outperforms conventional methods, such as Shallow Neural Networks (SNNs), Support Vector Regression (SVR), Bayesian networks. Later, Lai et al. (2018) proposed a Long- and Short-term Time-series Network (LSTNet) using the Convolution Neural Networks (CNN) and the Recurrent Neural Networks (RNN) to extract short-term local dependency patterns among variables and to discover long-term patterns for time series trends. A collection of 48-month hourly data from the California Department of Transportation was used to test the prediction accuracy. The results showed that the proposed LSTNet had an averaged improvement of 11.70% in RSE (Root Relative Squared Error) value compared with the baseline model RNN-GRU (Gated Recurrent Unit) and demonstrated the effectiveness of the proposed framework for complex repetitive patterns. Polson and Sokolov (2017) developed a deep learning architecture combining a linear model and a series of fully-connected layers to overcome the challenge of sharp non-linearities in traffic flow prediction. This proposed framework was tested using real-world traffic flow data under abnormal traffic conditions. The results showed that the prediction accuracy relatively improved by 14% compared to the conventional simple neural network model. Zhao et al. (2017) proposed a cascaded Long Short-Term Memory (LSTM) neural network considering temporal-spatial correlation in traffic networks. The experiments applied to three datasets collected from Beijing, China showed that compared with conventional neural network methods, the proposed LSTM network can achieve more accurate results with an averaged improvement of 6.22% in terms of mean relative error (MRE). Ma et al. (2017) converted the traffic dynamics to images in order to describe temporal-spatial correlations among links in a road network. The Convolutional Neural Networks (CNN) was applied to the converted images to learn multi-dimensional traffic data and predicted traffic speed at a large network-wide scale. The experiments reveal that the proposed method outperforms other methods by an average accuracy improvement of 42.91% in terms of Mean Squared Error (MSE).

Considering the problems of loss and irregularity in traffic flow datasets, Tian et al. (2018) applied a multi-scale temporal smoothing method to missing data imputation and a prediction method based on LSTM was proposed. Duan et al. (2018) proposed a hybrid neural network of CNNs and LSTM trained using real-world taxis GPS to capture the non-linear and spatial–temporal characteristics of urban traffic data. Later, Wei et al. (2019) combined the Autoencoder and LSTM method to firstly capture traffic flow features in upstream and downstream links via Autoencoder. The features were used as inputs in LSTM to provide 5-min interval multi-step traffic flow prediction. The experiments using traffic datasets collected from California PeMS system showed the forecasting error and fluctuation of the error were smaller than comparative models, such as CNN.

To capture higher dimensional temporal data patterns, the stacked LSTM network is widely used. The architecture and hyper-parameters of stacked LSTM network have been discussed by a number of researchers (Chen et al., 2017a; Cui et al., 2018;

Vinayakumar et al., 2017; Du et al., 2017; Shao and Soong, 2016). However, due to the specificity and variety between different datasets, there is not a universal LSTM architecture for all traffic datasets. For example, Vinayakumar et al. (2017) found that 6-layer stacked LSTM with 500 units in each layer performs best by experiments on sufficient 15-min-interval traffic data from GEANT networks whereas Du et al. (2017) demonstrated that 10-layer stacked LSTM is the most appropriate network for the 24-hour traffic count data in Dallas, Texas.

Although a large volume of literature in the area of short-term traffic prediction using machine learning tools exists, it was found that the necessary model calibration using sufficient data was quite time-consuming. Only few studies have investigated the transferability of machine-learning-based prediction models under the condition of insufficient data. The following subsection will focus on the review of model transferability in transportation.

### 2.2. *Previous studies of transferability in transportation*

In the last few decades, much work has been carried out using statistical methods in investigating the transferability of various transport models across time periods and different regions, e.g. travel demand models, route choice models (Bekhor and Prato, 2009) and accident prediction models (Hadayeghi et al., 2006). Brand and Cheslow (1979) defined transferrable models as models estimated with different datasets, which are invariant with regards to the estimated parameters. One main assumption made in their research is the identical distribution of values of the independent variables in the populations between which the models are being transferred (Brand and Cheslow, 1979).

There is, however, no widely-accepted definition of model transferability in machine learning. Surprisingly, little attention has been paid to the transferability of short-term traffic prediction models based on machine learning tools. Guo (2013) defined transferability as the ability of a machine learning model to work well without extensive site-specific calibration to capture the location-specific attributes. In Guo's research (2013), however, the re-calibration process of each location is still required. Luan et al. (2018) investigated the link-to-link transferability of three different machine learning models in the application of short-term travel time prediction. The good transferability of a trained machine learning model was defined as: the model prediction accuracy does not significantly decline after being transferred to the target link. The results showed that the road capacity and traffic demand are the main factors to influence the performance of transfer between links. However, in Luan et al.'s reaearch, the models trained from the source link dataset are directly applied to the target link without any modification, which leads to huge prediction error in some cases. Accordingly, it is significant to adjust the established model towards existing information in the target link to overcome the data difference between the source link and target link. In this paper, transferability is defined as the ability of a machine learning model to work well without extensive parameter and layer re-calibration across different locations and sites.

As reviewed previously, a method that has been demonstrated to properly address the challenge of transferability is to apply transfer learning to Deep Neural Networks (DNN) to capture more robust features in datasets and improve model transferability (Pan and Yang, 2009).

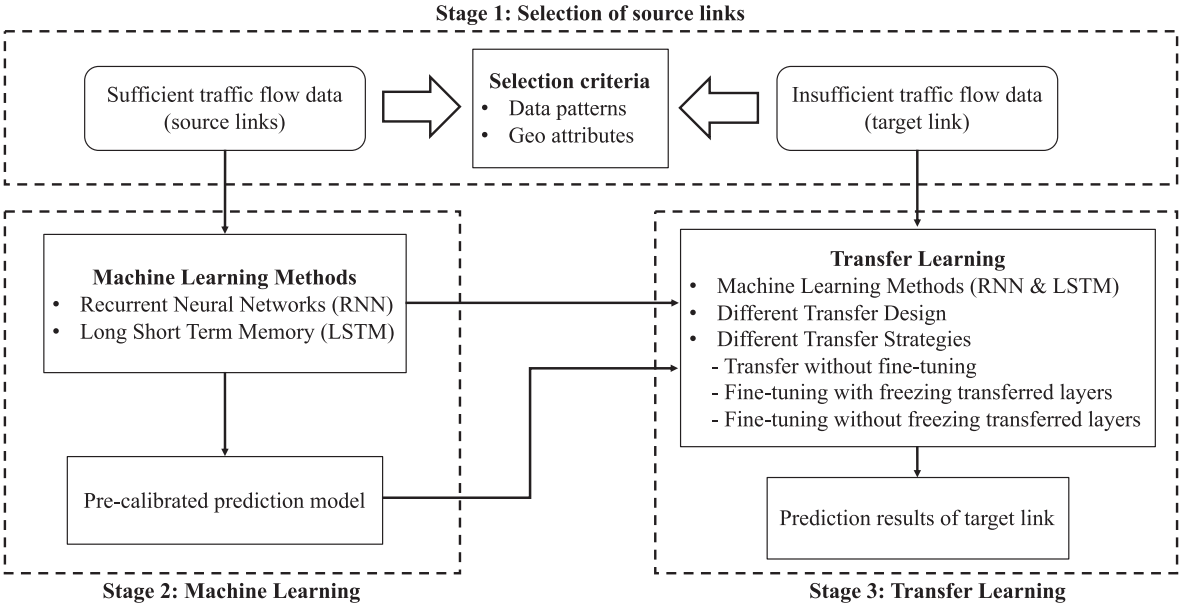



**Fig. 1.** Flowchart of the proposed 3-stage prediction framework.

## 3. Research methodology

### 3.1. Prediction framework

In order to address insufficient data problems, this section proposes a general 3-stage traffic prediction framework with transfer learning to improve the transferability of machine-learning-based models of short-term traffic prediction. The proposed 3-stage prediction framework is illustrated in Fig. 1. In order to predict the future values of a target link with insufficient historical data, a set of source links with sufficient historical data should be selected in Stage 1. The link geographical attributes-based data selection criterion, which has been demonstrated in Luan et al. (2018), is applied in this paper. Additionally, the data patterns between links are also considered. One of the objectives of this paper is to identify the impacts of different input attributes (i.e., different selection criteria). A series of scenarios will be designed in the next section. In Stage 2, machine learning methods are applied to train the pre-calibrated model using sufficient historical data of a group of source links. Finally, in Stage 3, some layer-parameters in the pre-calibrated model are transferred to create a new network using insufficient data from the target link.

**Stage 1: Selection of source links**

In the first stage, the main objective is to identify the selection criteria among traffic datasets. When the target link suffers from the insufficient data problem, the approach to find suitable source datasets with sufficient data that can be used to calibrate a transferable prediction model to the target link should be determined.

**Stage 2: Machine Learning**

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) as typical Deep Neural Networks (DNN) are selected in Stage 2 of the proposed framework. Both have been widely used in the existing research and applications to tackle time series problems in traffic prediction domain (Zhao et al., 2017; Lai et al., 2018; Ma et al., 2015). In this paper, the LSTM is used to avoid the vanishing or exploding gradients problem of long-sequence RNN model.

*Long Short-Term Memory (LSTM)*

The Long Short-Term Memory (LSTM) cell was firstly proposed by Hochreiter and Schmidhuber (1997), and it was gradually improved over the years (Sak et al., 2014; Zaremba et al., 2014). To solve the vanishing or exploding gradients problem of long-sequence RNN model in the back-propagation process, its state is split into two vectors: a hidden state $h_{(t)}$ to determine the short-term state and a cell state $c_{(t)}$ to determine the long-term state. Due to its capability in capturing temporal data patterns, LSTM has been widely applied in traffic prediction (Cui et al., 2018; Vinayakumar et al., 2017; Du et al., 2017).

*Stacked LSTM*

Although one single LSTM cell is able to solve the vanishing or exploding gradients problem of long-sequence RNN model, its prediction accuracy is still limited by the simple network structure. As a result, the stacked LSTM, where multiple layers of LSTM are placed over each other and both the long-term state $c_{(t)}$ and the short-term state $h_{(t)}$ are propagated within LSTM cells, is applied in this paper to capture the high-dimensional non-linear relationships in datasets.

**Stage 3: Transfer learning**

Given the definition in Pan and Yang (2009), the concepts of a domain and a task are defined in transfer learning. A domain $\mathscr{D}$ is comprised of a feature space $\mathscr{X}$ and a marginal probability distribution $P(X)$ over the feature space. The relationship between $\mathscr{X}$ and $X$ can be denoted as $X = \{x_1, \cdots, x_n\} \in \mathscr{X}$, where $X$ is a particular learning sample, $\mathscr{X}$ is a space containing all features, $x_i$ is the $i^{th}$ vector in the existing learning sample $X$ describing different features. Given a domain $\mathscr{D} = \{\mathscr{X}, P(X)\}$,a task $T$ is comprised of a label space $Y$ containing all labels $y_i$ corresponding to learning vectors $x_i$ and a conditional probability distribution $P(Y|X)$ (i.e. the objective predictive function $f(\cdot)$) that is typically learned from the training pairs $x_i \in X$ and $y_i \in Y$.

Given a source domain $\mathscr{D}_S$, a corresponding source task $T_S$, as well as a target domain $\mathscr{D}_T$ and a target task $T_T$, transfer learning aims to enhance the learning of the target conditional probability distribution $P(Y_T|X_T)$ (i.e. the objective predictive function $f(\cdot)$) in $T_T$ with the knowledge learned from $\mathscr{D}_S$ and $T_S$, where $\mathscr{D}_S \neq \mathscr{D}_T$ or $T_S \neq T_T$ (Pan and Yang, 2009). More details of transfer learning can be found in Pan and Yang (2009).

The concept of neural network based transfer learning method involves several notions on $\{\mathscr{D}_S, T_S\}$ and $\{\mathscr{D}_T, T_T\}$, which are defined as follows:

- *A source dataset A*: denoted by $\mathscr{D}_S = \{X_S, P(X_S)\}$, where the learning samples in $X_S$ are sufficient enough that contains almost all the features in the feature space $\mathscr{X}$ and the distribution $P(X)$ over the feature space.
- *A source task A:* denoted by $T_S = \{Y_S, f_S(\cdot)\}$, where the objective predictive function $f_S(\cdot)$ can be accurately estimated via sufficient training pairs $x_i \in X_S$ and $y_i \in Y_S$.
- *A source network A*: denoted by $f_S(\cdot)$, where all the parameters within the network are precisely estimated to enable $f_S(\cdot)$ providing accurate predictions.
- *A target dataset B*: denoted by $\mathscr{D}_T = \{X_T, P(X_T)\}$, where the learning samples in $X_T$ are very limited that $\{\mathscr{X}, P(X)\}$ cannot be represented by $\{X_T, P(X_T)\}$
- *A target task B:* denoted by $T_T = \{Y_T, f_T(\cdot)\}$, where the objective predictive function $f_T(\cdot)$ cannot be accurately estimated due to insufficient training pairs $x_i \in X_S$ and $y_i \in Y_S$.
- *A target network B*: denoted by $f_T(\cdot)$, where most of the parameters within the network are not precisely estimated so that the prediction accuracy can't be guaranteed.

Given the definitions above, three types of transfer strategies in DNN-based transfer learning method, namely transfer without fine-tuning, fine-tuning with freezing transferred layers and fine-tuning without freezing transferred layers (Stanford University, 2019), are used in this paper and illustrated as below.

The simplest transfer strategy is to treat the entire *source network A* as a fixed feature extractor without any further layer-parameter adjustments (i.e. transfer without fine-tuning). In this strategy, the *source network A* is copied and directly transferred toward the *target dataset B*. This simple strategy has been used in the applications of image processing and traffic prediction. For example, Donahue et al. (2014) validated that a generic visual feature extracted from the entire ImageNet-pretrained ConvNet outperforms conventional visual representations extracted by benchmark methods. Similarly, Sharif Razavian et al. (2014) extracted features from the ImageNet-pretrained ConvNet and directly re-purposed these features to train an SVM model to deal with novel generic tasks. More recently, Luan et al. (2018) investigated the link-to-link transferability of different machine learning models in short-term travel time prediction by transferring the entire pre-trained model to the target link without any re-calibration.

However, due to differences between *source dataset A* and *target dataset B*, the features extracted by the *source network A* might be specific to the *source task A* but not generic to the *target task B*. As a result, the fine-tuning method is necessary to adjust the transferred network layout toward the *target task B* to overcome the differences between A and B. The basic idea of the fine-tuning strategy is to use different layer-parameter adjusting methods to fine-tune a transferred network, where the first $n$ layers of the transferred network are copied from the *source network A* and the remaining higher layers are initialized randomly. There are two types of fine-tuning strategies:

- Fine-tuning with freezing transferred layers: parameters in the first $n$ layers are frozen, and parameters in higher layers are trained toward *target dataset B*.
- Fine-tuning without freezing transferred layers: all the layer-parameters within the full network are retrained toward *target dataset B*.

The main factors used to select fine-tuning strategies are the complexity of the transferred network, the size of the *target dataset B*, and its similarity to the *source dataset A*. Yosinski et al. (2014) investigated the two fine-tuning strategies in image recognition and showed that the choice of fine-tuning strategies depends on the size of the target dataset and the number of parameters in the first $n$ layers. Fine-tuning with freezing transferred layers is usually used in the condition of the small size of *target dataset B*. In this situation, data in $B$ is insufficient to re-train the full network but sufficient to fine-tune the higher-level portion of the network.

Moreover, Yosinski et al. (2014) also investigated the sensitivity of the number of transferred layers ($n$) and quantified the transfer performance improvements due to transferred features. The results showed that transferred features from the first 3 layers of source network are applicable to the target dataset. In this paper, based on the experiments in Yosinski et al. (2014), a similar sensitivity analysis toward the transfer strategies and the number of transferred layers ($n$) is conducted in the next section to discuss not only the transferred features but also the overfitting issues.

### 3.2. Evaluation criteria of prediction accuracy

The prediction accuracy is evaluated using both scale-dependent criteria (Mean Squared Error (MSE), Mean Absolute Error (MAE)) and scale-independent criteria (Mean Absolute Percentage Error (MAPE)). To evaluate the transferability of a machine learning-based model between different datasets, the MAPE, which is a unit independent goodness-of-fit measurement, is the main focus in this research. The three measures are calculated as follows:

Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \widehat{Y}_i\right| \tag{1}$$

Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2 \tag{2}$$

Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Y_i - \widehat{Y}_i}{Y_i}\right| \tag{3}$$

where $Y_i$ represents the traffic observations, $\widehat{Y}_i$ is the traffic predictions, $n$ represents the total number of traffic observations.

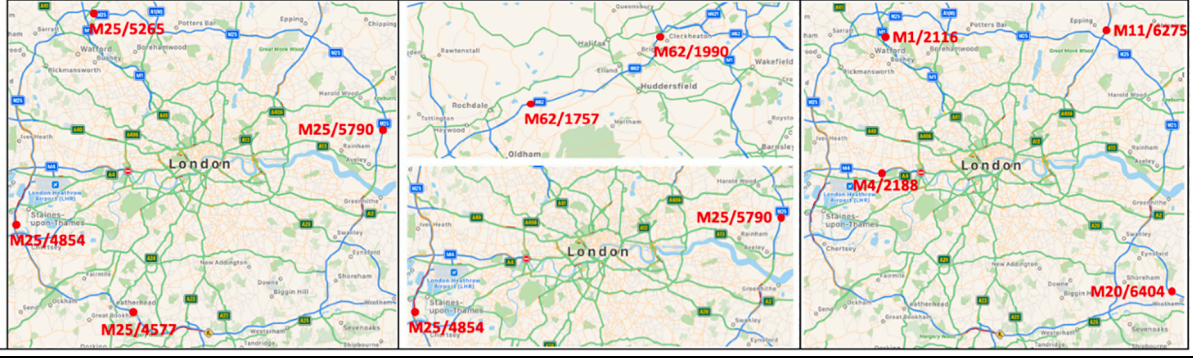
## 4. Empirical analysis

### 4.1. Problem formulation

To make the algorithm fully learn the relationships within traffic data, the original one-dimensional time series is transferred to a

**Table 1**
Details of selected links in three scenarios.

| | Scenario 1 | | | | Scenario 2 | | | | Scenario 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Source links | | | Target link | Source links | | | Target link | Source links | | | Target link |
| | M25-5790 | M25-4854 | M25-5265 | M25-4577 | M62-1757 | M62-1990 | M25-5790 | M25-4854 | M11-6275 | M1-2116 | M4-2188 | M20-6404 |
| Average Cross-correlation Coefficient | 0.9723 | | | | 0.9635 | | | | 0.8541 | | | |
| Speed Limit | 70 mph | | | | 70 mph | | | | 70 mph | | | |
| Number of Lanes | 4 | 5 | 4 | 4 | 3 | 4 | 4 | 5 | 4 | | | |
| Direction | Orbital | | | | Inbound | Outbound | Orbital | Orbital | Inbound | | | |
| Emergency Stop Lane | Yes | | | | Yes | No | Yes | Yes | Yes | | | |
| Near ramps (<300m) | No | | | | Yes | No | No | No | No | | | |
| Location | London | | | | Manchester | Leeds | London | London | London | | | |



multi-dimensional series, which forms the trajectory matrix:

$$T_X = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_K \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & \cdots & x_L \\ x_2 & x_3 & \cdots & x_{L+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_K & x_{K+1} & \cdots & x_{K+L-1} \end{pmatrix} \tag{4}$$

where $L$ is the window length; $N = K + L - 1$, $N$ is the length of the original one-dimensional time series. The window length is set to be 5 according to the sensitivity results in previous sensitivity studies (Pascale and Nicoli, 2011; Guo et al., 2012). The training dataset is split into two parts, the input trajectory matrix $T_{Input}$ and the label vector$T_{Label}$:

$$T_{input} = \begin{pmatrix} x_1 & x_2 & \cdots & x_5 \\ x_2 & x_3 & \cdots & x_6 \\ \vdots & \vdots & \ddots & \vdots \\ x_K & x_{K+1} & \cdots & x_{K+4} \end{pmatrix}, \; T_{Label} = \begin{pmatrix} x_6 \\ x_7 \\ \vdots \\ x_{K+5} \end{pmatrix} \tag{5}$$

Two types of factors are investigated in this paper, namely traffic flow patterns of links and geographical attributes of each link. Given the feature space $\mathscr{X}$, the marginal probability distribution $P(X)$ is denoted as both the data patterns of links (e.g. data distribution, traffic demands) and a wide array of combinations and variations in link geographical attributes (e.g. numbers of lanes, speed limits, location of links).

Given a defined domain $\mathscr{D} = \{\mathscr{X}, P(X)\}$ as described above, a task $T$ consists of a label space $Y$, which includes the label vector $T_{Label}$, and a conditional probability distribution $P(Y|X)$ (i.e. the objective predictive function $f(\cdot)$) that is typically learned from the training pairs $x_i \in T_{input}$ and $y_i \in T_{Label}$.

Therefore, the transfer learning problem in this paper is defined as finding an optimised approach to help improve the learning of the target predictive function $f_T(\cdot)$ using the knowledge in $\mathscr{D}_S$ and $T_S$, where $\mathscr{D}_S = \mathscr{D}_T, T_S \neq T_T$. Specifically, $T_S \neq T_T$ involves different conditional probability distribution $P(Y|X)$ (i.e. the objective predictive function $f(\cdot)$) between different links. The subscript $S$ represents source links and $T$ means target link.

Due to the uncertainty of traffic flow, the inevitable noise and various geographical attributes, it is impossible to find links which strictly follow the same distribution and possess the same geographical attributes. The strict conditions for both traffic flow patterns and geographical attributes need to be relaxed. To quantify data patterns between different links, the cross-correlation coefficient is used to judge data consistency between links, which is formulated as:

$$r(d) = \frac{\sum_i \left[ \left( x_{(i)} - M_{(x)} \right) * \left( y_{(i-d)} - M_{(y)} \right) \right]}{\sqrt{\sum_i \left( x_{(i)} - M_{(x)} \right)^2} * \sqrt{\sum_i \left( y_{(i-d)} - M_{(x)} \right)^2}} \tag{6}$$

where $x_{(i)}$ and $y_{(i)}$ are two series, $i = 0,1,2,\cdots,N-1$; $M_{(x)}$and $M_{(y)}$ are the means of the corresponding series and $d$ represents the delays, $d = 0,1,2,\cdots,N-1$.

In order to investigate the transferability of machine-learning-based prediction models, three scenarios have been designed based on traffic flow patterns (denoted as the cross-correlation coefficient) and link geographical attributes (denoted as combinations and variations of numbers of lanes, speed limits, location of links, etc.):

- **Scenario 1:** Transferring links with high cross-correlation coefficient (>0.95) and highly consistent geographical attributes
- **Scenario 2:** Transferring links with high cross-correlation coefficient (>0.95) and roughly similar geographical attributes
- **Scenario 3:** Transferring links with medium cross-correlation coefficient (>0.8) and highly consistent geographical attributes

### 4.2. Traffic data

The traffic flow data used in this paper is obtained from the Highways England (http://webtris.highwaysengland.co.uk/). All the traffic data used in this paper is 15-minute traffic flow data collected from Inductive Loop Detectors (ILDs). In addition, some geographical attributes data such as speed limit, number of lanes and location of ramps are collected through Google Maps (www.google.com/maps) and OpenStreetMap (www.openstreetmap.org).

Traffic data used in three scenarios according to their data consistency and geographical attributes are summarized in Table 1. In three scenarios, the target links, the sites M25-4577, M25-4854 and M20-6404, are randomly chosen and the rest links are source links in each scenario. For all the source links, the training data were collected from 1st January to 15th Aug 2019. For each target link, only 3-day traffic flow data was selected from 19th Aug 2019 to 21st Aug 2019 for training to create the insufficient training datasets and the following 10-day data from 22nd Aug 2019 and 31st Aug 2019 was selected for testing.

### 4.3. Hyper-parameter selection of baseline networks

In order to maximise the efficacy of stacked LSTM structure and avoid overfitting problems, a number of experiments are firstly conducted on Scenario 1 Database (i.e., M25-5790, M25-4854, M25-5265, M25-4577). The optimal architecture of LSTM and its

parameters were identified through grid search method in this paper (Bergstra and Bengio, 2012). Grid search, as one of the most widely used methods to optimise hyper-parameters of machine learning, is a process that could search the optimised parameters through a manually specified subset (Bergstra and Bengio, 2012). Based on trials and errors, we empirically found that the network with 3 stacked LSTM layers with 16 units in each hidden layer is the most appropriate for the *source dataset A*, this network is denoted as *source network A* and yields a promising prediction accuracy in task A (6.87% MAPE). However, when trained and tested on the *target dataset B*, the prediction accuracy of this network structure (denoted as *target baseline network B1* here) is 11.65% in terms of MAPE, due to the high model complexity and limited amount of data. A similar hyper-parameter sensitivity analysis is conducted on *target dataset B* and the simple 1-layer LSTM network with 4 units in hidden layer is empirically selected (denoted as *target baseline network B2* and accomplished 9.32% MAPE). In addition, it is worth noting that an additional fully connected (FC) layer is added after the stacked LSTM layers to reshape and linearly activate the output of the last LSTM layer. The chosen network architecture and examples of their 15-minute ahead prediction results are shown in Figs. 2 and 3.

Due to sufficient high-quality traffic flow data as well as the optimized network structure, the network A outperforms the others in all situations. It is important to note that although network B2 shows an acceptable prediction performance on weekdays, it is not well predicted during weekends. The data patterns between weekdays and weekends are quite different. Due to the limited data quantity and model size, network B2 can only learn from the weekday data but will never capture traffic flow features on the weekends. What's worse, some abnormal features during weekdays that don't exist in the limited training dataset will also be ignored in the case of network B2, leading to a series of inaccurate and non-robust prediction results.

### 4.4. Hyper-parameter selection of transferred network

Given the selection and performance of the baseline networks above, Scenario 1 is also used to analyse hyper-parameter sensitivity of DNN based transfer learning methods. The *source network A* and the *target baseline network B1* are the focus in this section when we investigate the layer-parameter transferability, and the *target baseline network B2* is only used as a control group.

There are five key design hyper-parameters when applying DNN based transfer learning approaches to short-term traffic prediction:

- Number of transferred layers
- Transfer strategies
- Choice of learning rate
- Choice of batch size
- Choice of training epoch

***Number of transferred layers***

In transfer learning, the first *n* layers of the transfer network are copied from *source network A* and the remaining higher order layers

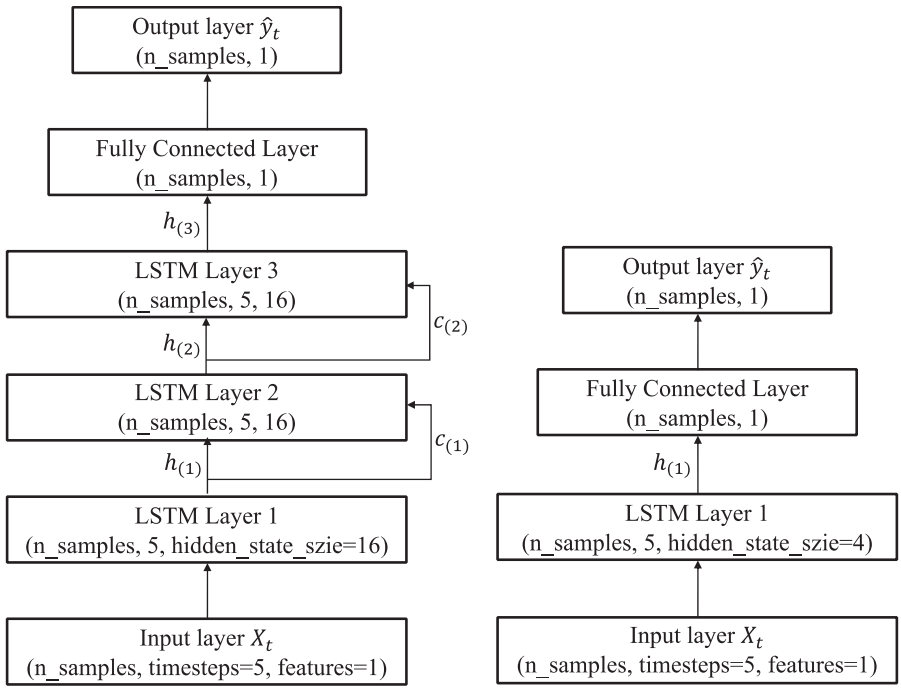



**Fig. 2.** Architectures of Baseline Networks (left: Network A & B1, right: Network B2).

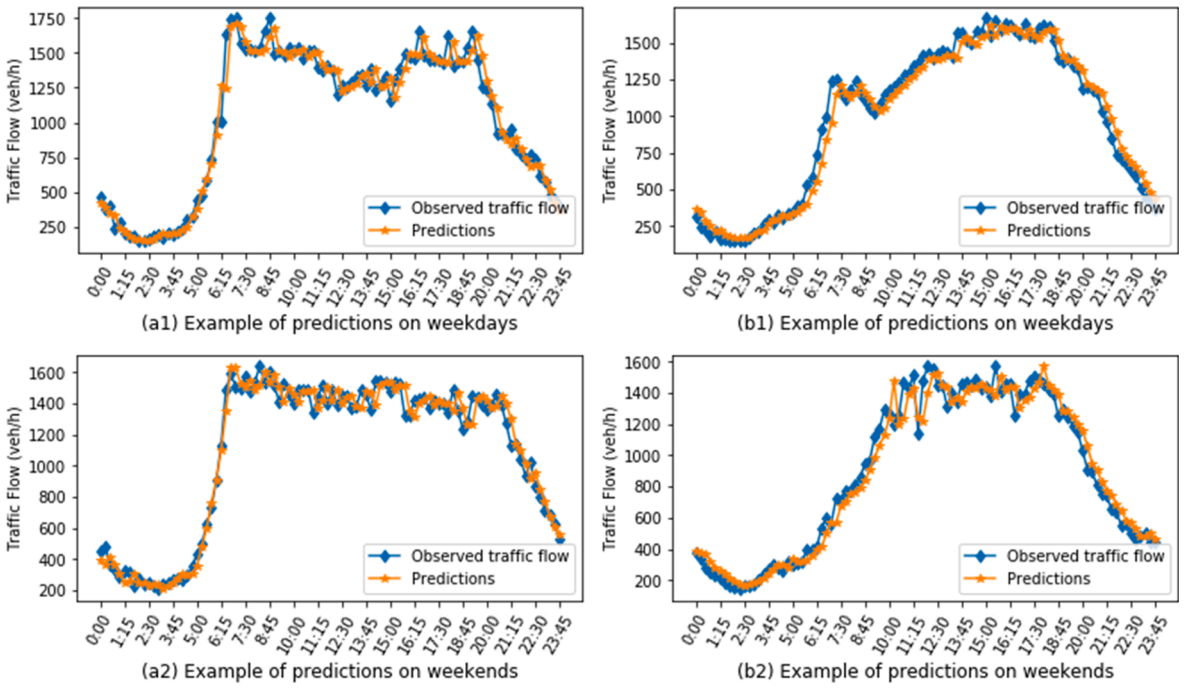



**Fig. 3.** Examples of Prediction Results (left: *source network A*, right: *baseline network B2*).

$(4-n)$ are initialized randomly. Intuitively, if the transferred model has more accurate results than (or as good as) the *target baseline network B1*, there is evidence that the features of the first *n*-layers are general and can be transferred to solve serious data insufficiency problems in short-term traffic prediction. If not, there is evidence that the first *n* layers of *source network A* have low transferability and this hybrid method is ineffective at the *n*-layer transfer level.

***Transfer strategies***

In practice, the pre-trained *source network A* is used either as an initialisation or a fixed feature extractor for the task of interest. Specifically, according to different layer-parameter adjusting techniques in the transferred network, the three main transfer learning strategies are as follows:

- **Strategy 1:** Fixed feature extractor (transfer without fine-tuning): parameters in first *n* layers remain unchanged, and the randomly initialized parameters in higher order layers $(4-n)$ are replaced by trained parameters which are directly extracted from the higher order layers $(4-n)$ in *target network B1*.
- **Strategy 2:** Fine-tuning with freezing transferred layers: parameters in first *n* layers are frozen, and parameters in higher order layers $(4-n)$ are trained toward *target dataset B*.
- **Strategy 3:** Fine-tuning without freezing transferred layers: parameters within the entire network are retrained toward *target dataset B*.

Particularly, if $n = 4$, the entire new network will be completely formed by 4 layers copied and transferred from the *source network A*. In this case, fine-tuning in Strategy 2 is no longer useful as there are no adjustable parameters to be trained by the *target dataset B*. Hence, an additional fully connected (FC) layer was added after the 4 frozen transferred layers to create a new network with 5 layers, which is trained by the insufficient *target dataset B*. This newly created 5-layer network is denoted as $n = 4 + 1$ and also applied to Strategy 3.

***Choice of learning rate***

Learning rate refers to the amount that the parameters are updated during training, which involves a trade-off between the speed of convergence and accuracy of convergence. Depending on how the learning rates are calculated at each step, optimisation methods can be grouped into two categories, non-adaptive learning rate methods, such as stochastic gradient descent (SGD), SDG with momentum, and adaptive learning rate methods including AdaGrad (Duchi et al., 2011), RMSProp (Tieleman and Hinton, 2012) and Adam (Kingma and Ba, 2014). Adaptive learning rate methods have become increasingly popular due to their rapid training time and good performance (Karparthy, 2017). In this paper, the RMSProp is chosen for two reasons, firstly, it runs faster than the non-adaptive learning rate methods; secondly, it implements an exponential decay function to the historical squared sum of gradients to resolve the issues of reducing learning rates too quickly as in other adaptive methods, such as AdaGrad which would result in the optimisation never converging to the global minimum, or poor performance in non-convex situations (Géron, 2019). The adaptive learning rate $\eta$ using the RMSProp optimizer is selected as $\eta = \eta_0 / \sqrt{E[g^2]_t + \epsilon}$, where $\eta_0$ is the initial learning rate (set to 0.001 as the commonly used

value), $E[g^2]_t$ represents the moving average of squared gradients, $E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma)g^2{}_t$, $\gamma$ is the decay rate (i.e., typically set to 0.9 by default), $g_t$ is the gradient of the cost function with respect to the weight, and $\in$ is a constant to avoid the denominator being zero ($10^{-8}$ by default). At each training step, the learning rate $\eta$ is kept updated based on the rooted mean square of past gradients with decay.

***Choice of batch size***

Batch size refers to the number of examples from the training dataset used in the estimate of the error gradient, which is an important hyper-parameter that influences the dynamics of the learning algorithm. The relationship between batch size, convergence speed and accuracy has been discussed in some studies, such as Radiuk (2017) and Keskar et al. (2016). However, the choice of batch size is closely related to data size and the number of features. Mini batch training is usually used when there are a large number of training instances, as a smaller batch size would result in more frequent gradient updates due to reduced computational costs. However, it would also lead to more erratic progress in parameter space because only a sample of the training data is visible to the gradient updating process (Géron, 2019). According to Hinton et al. (2012), the *full batch learning* strategy is effective for small datasets (e.g. 10,000 cases) for its capacity in avoiding the mini-batch sampling error. As the amount of training data is very limited under the scenario design, the *full batch learning*, where all the training pairs $x_i \in T_{input}$ and $y_i \in T_{Label}$ are included in one batch, is applied in this paper to guarantee a more accurate gradient descent.

***Choice of training epoch***

The number of training epoch demonstrates how many times the entire dataset is fed through a neural network. There is not a universal optimum number of training epochs and it is highly dependent on the size and diversity of the training dataset. Moreover, the overfitting problem is more significant in our transfer learning empirical studies since only three-day traffic flow data is assumed to be available in the target dataset. Hence, we applied the *early stopping* strategy to avoid the overfitting problem. A training epoch sensitivity analysis example is also conducted to further discuss the efficacy of our transfer learning method.

To clearly demonstrate the architecture of the three transfer strategies, an example with 3 transferred layers is shown in Fig. 4.

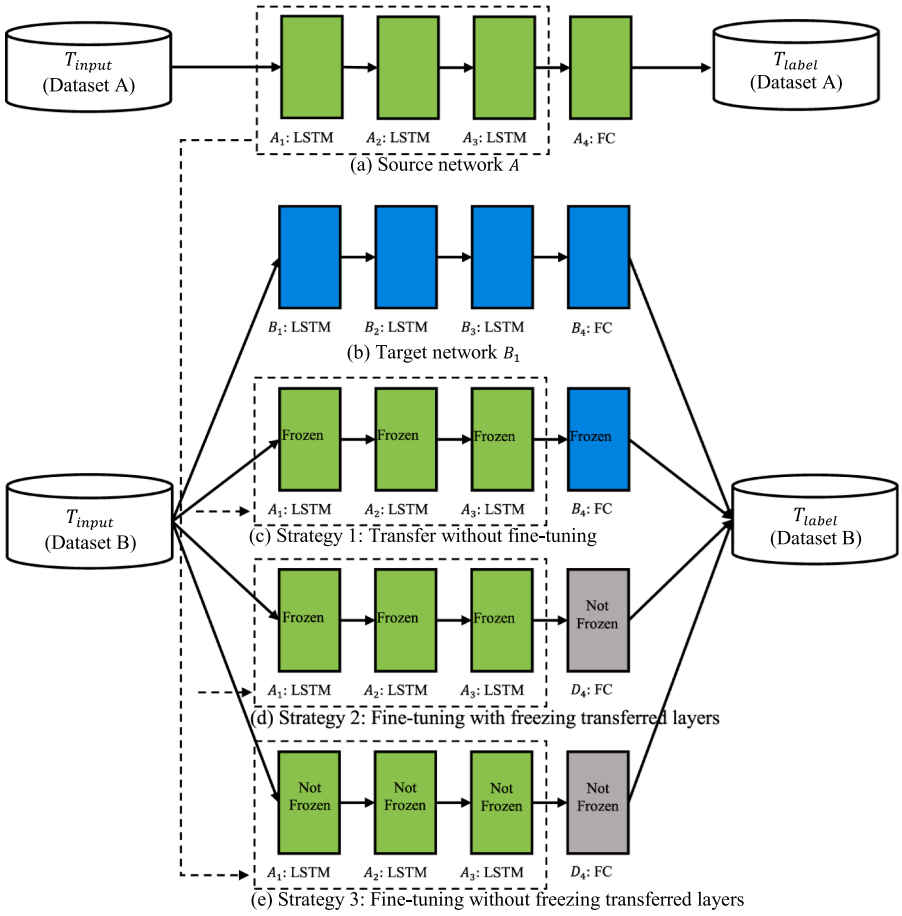



**Fig. 4.** Example of three cross transfer strategies with 3 transferred layers.

### 4.5. Experimental results of hyper-parameter selection

A series of experiments have been designed to investigate the transferability of features from each layer of the *source network A* and test the sensitivity of the main hyper-parameters in transfer learning. The accuracy results of different transfer strategies and designs are shown in Table 2 and Fig. 5.

It can be seen in Table 2 that the model using 4 transferred layers from *source network A* with one extra layer created using *target dataset B* has the most accurate prediction results. The results show that traffic prediction in short-term using transfer learning with fine-tuning methods outperform the method without fine-tuning no matter the number of transferred layers. Apart from Baseline Net B1 & B2, two naïve prediction approaches (i.e. Historical Average (HA)) are applied in our experiments for comparisons, where HA1 calculates an average per slot of the day and HA2 uses the rolling mean of the last several values as the prediction. A T-test between transfer Strategy 2 (4 + 1) and Baseline Net B2 is conducted and the results show that the prediction accuracy difference is statistically significant at 95% confidence level (p-value equals 0.0075). In the fine-tuning strategies (Strategy 2 and 3), Strategy 2 (fine-tuning with freezing transferred layers) has more accurate prediction results than those in Strategy 3 (fine-tuning without freezing transferred layers). This is because under such a hypothetical data missing scenario (only 3-day data is available), the data is insufficient to re-train the full network but sufficient to fine-tune the higher-level portion of the network, thus the overfitting issue always dominant in the unsatisfactory performance in Strategy 3.

Theoretically, the performance improvement of the transferred network is mainly brought by the alleviation of overfitting and the transferred features extracted from *source dataset A*. Under this assumption, the two main factors influencing transfer efficacy are quantified in Fig. 5.

Fig. 5(b) shows the comparisons between Transfer Strategy 2 and two baseline networks. The whole blue area between the blue line and the dash line represents the improvements of Transfer Strategy 2 to the baseline network B1, which results from both alleviation of overfitting and transferred features. It is worth to note that although Transfer Strategy 2 is able to improve the prediction accuracy from the two aspects above, overfitting issue also exists in this transferred network. Compared with the *baseline network B2* (the optimized network without overfitting problems), only the network with 3 or more transferred layers outperforms network B2, which indicates that at least 3 layers from the *source network A* is required to recover the performance degradation brought by the overfitting problem and lead to an overall performance improvement (shown as dark blue area in Fig. 5(b)). Based on the discussions above, we can theoretically explain the incomparable performance of the '4 + 1' group in Transfer Strategy 2 from the two aspects:

1. All the features extracted from *source dataset A* are transferred to the *target network B* in a layer-parameter form, which enables *target network B* to identify traffic flow patterns that do not exist in *target dataset B*.
2. Only one additional layer after the 4 freezing transferred layers is created to fine-tune the whole network toward *target dataset B*. Hence, the difference between two datasets can be overcome and little overfitting issue will occur as given the existing *target dataset B,* 1-layer network has been proven to be the most effective and efficient in the previous hyperparameter sensitivity analysis.

Similarly, in Fig. 5(c), the blue area shows the prediction accuracy improvement brought by transferred features only, since the two networks suffer from the same overfitting issue when training toward *target dataset B*. Compared with *baseline network B2*, the red area shows that even though the prediction accuracy of transferred network is getting better, the overall overfitting problem is always dominant in the transferred network and is never totally recovered by the transferred features. Especially, the red line from 3-layer on

**Table 2**
Results of hyper-parameter sensitivity experiment.

| Strategies | Evaluation Criteria | Number of Transferred Layers | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0 (Network $B_1$) | 1 | 2 | 3 | 4 | 4 + 1 |
| Strategy 1: Transfer without Fine-tuning | MSE | 12890.31 | 30600.56 | 42970.34 | 37315.76 | 8756.98 | – |
| | MAE | 83.35 | 136.77 | 177.32 | 162.99 | 65.93 | |
| | MAPE | 11.65% | 24.14% | 41.32% | 35.71% | 9.16% | |
| Strategy 2: Fine-tuning with Freezing Transferred Layers | MSE | 12890.31 | 9467.6 | 8389.41 | 8294.03 | 8756.98 | 8463.64 |
| | MAE | 83.35 | 71.09 | 65.79 | 63.41 | 65.93 | 60.41 |
| | MAPE | 11.65% | 10.95% | 9.69% | 8.63% | 9.16% | 7.91% |
| Strategy 3: Fine-tuning without Freezing Transferred Layers | MSE | 12890.31 | 9540.25 | 9355.02 | 9112.26 | 9057.82 | 9210.02 |
| | MAE | 83.35 | 71.41 | 70.55 | 67.85 | 64.05 | 69.24 |
| | MAPE | 11.65% | 11.02% | 10.84% | 9.59% | 9.65% | 10.07% |
| Baseline Network $B_2$ | MSE | 8640.5 | | | | | |
| | MAE | 65.69 | | | | | |
| | MAPE | 9.32% | | | | | |
| HA1 | MSE | 65966.77 | | | | | |
| | MAE | 209.52 | | | | | |
| | MAPE | 23.77% | | | | | |
| HA2 | MSE | 9156.69 | | | | | |
| | MAE | 69.54 | | | | | |
| | MAPE | 10.44% | | | | | |

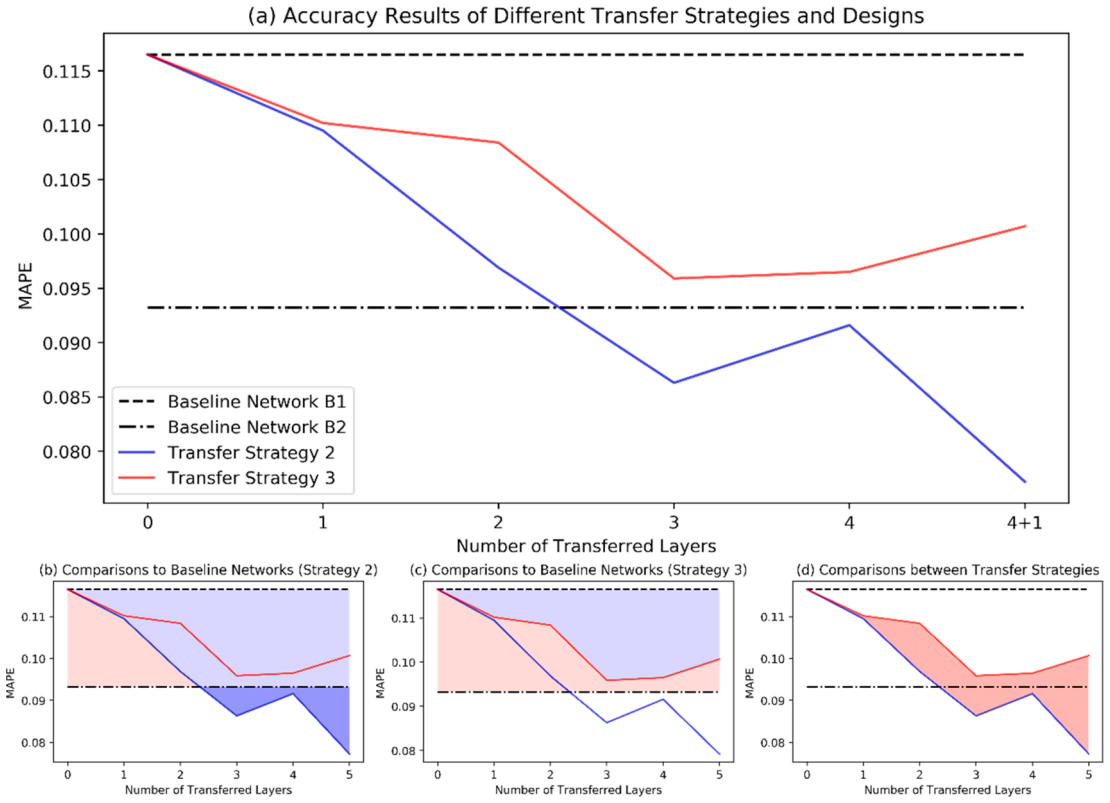



**Fig. 5.** Results of Hyper-parameter Sensitivity Experiment.

indicates a more severe overfitting issue that even can't be recovered by newly transferred features and finally results in the overall performance drop.

Finally, the red area in Fig. 5(d) represents the prediction accuracy difference between Transfer Strategy 2 & 3, which is mainly due to different levels of overfitting problems and transferred features in the two strategies. Compared with Transfer Strategy 2, Strategy 3 suffers from more severe overfitting problems as the whole network needs to be retrained toward *target dataset B* whereas only the higher-level portion of the network needs to be retained in Transfer Strategy 2. What is worse, the transferred information in the long-term state $c_{(t)}$ and short-term state $h_{(t)}$ is likely to get forgotten during the retraining process of Strategy 3, inevitably leading to significant transfer performance degradation.

The effects of the training epoch are shown in Table 3 and Fig. 6. Obviously, due to the knowledge transfer from the source network, both transfer learning strategies are able to significantly reduce the model training time and provide better prediction accuracy. The results show that the Baseline Net B2 (training from scratch) does not converge even at 275 training epochs whereas both transfer learning models can converge within 175 epochs. This demonstrates the efficacy of our proposed model for rapid adaptation in target tasks. In addition, compared with Strategy 2, Strategy 3 converges more rapidly since all of its layer parameters are effectively initialized whereas Strategy 2 needs to train its additional fully connected (FC) layer from scratch. However, the overfitting problem is more significant in Strategy 3 when training epochs get lager. This is because this non-frozen transfer model contains a lot of parameters to be trained towards the limited target dataset, which is too complex relative to the amount of representative training data on the target link (like the Baseline Net B1). In this case, the transferred knowledge is likely to decay since these layer-parameters are repetitively adjusted to fit the limited target dataset. This problem is less obvious in Strategy 2 because all the transferred layer-parameters are frozen thus the transferred knowledge is reserved and only the last-layer parameters are adjusted towards the

**Table 3**
An example of training epoch sensitivity analysis.

| Strategy | Number of training epoches | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 50 | 75 | 100 | 125 | 150 | 175 | 200 | 225 | 250 | 275 |
| Strategy 2 (4 + 1) | 35.64% | 18.71% | 13.42% | 11.22% | 9.32% | 8.13% | 7.97% | 8.08% | 8.22% | 8.17% |
| Strategy 3 (4) | 9.41% | 9.37% | 9.55% | 9.63% | 9.91% | 10.18% | 10.33% | 10.54% | 10.61% | 10.77% |
| Baseline Net $B_2$ | 66.48% | 66.26% | 65.80% | 56.05% | 42.99% | 29.39% | 22.62% | 18.77% | 15.52% | 12.76% |

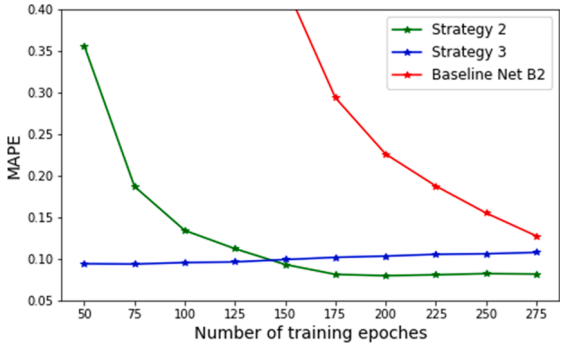



**Fig. 6.** Effects of training epoch.

target dataset. In this case, the overfitting problem is effectively controlled. These findings and conclusions are consistent with the applications of transfer learning techniques in the image classification area (e.g., Yosinski et al. (2014), Stanford University. (2019)).

Based on the results in the optimisation process, the design hyper-parameters of transfer learning used between links with high cross-correlation coefficient and highly consistent geographical attributes are as follows.

- Transfer strategy: Fine-tuning with freezing transferred layers
- Number of transferred layers: 4 + 1
- Batch size: Full Batch Learning
- Learning rate: Adaptive via RMSProp optimiser.
- Training epoch: Early stopping strategy

### 4.6. Scenario results

The comparisons of prediction accuracy in three scenarios with and without transfer learning are given in Table 4. Only MAPE measurement is used to make comparisons because the absolute traffic flow volume on different links are completely different. Hence, it is meaningless to make comparisons with scale-dependent criteria (MAE and MSE). In each scenario, both one-step ahead and multi-step ahead are tested to investigate the capability of transfer learning in short-term traffic prediction. The prediction horizon range is from 15 min (1-step) to one hour (4-step). The T-test is conducted to test the statistical significance between the proposed transfer learning method and Baseline Net B2 in 1-step prediction of each scenario. The results show that the prediction accuracy difference is statistically significant in Scenario 1 & 2 at 95% confidence level (p-value equals 0.0075 and 0.0194 respectively) but insignificant in

**Table 4**
Comparison of prediction accuracy in three scenarios.

| Experiment scenario | | Prediction with HA1 | Prediction with HA2 | Prediction without Transfer Learning (Baseline Network B2) | Prediction with Transfer Learning (Transfer Strategy 2) | Improvement to Baseline Net B2 |
|---|---|---|---|---|---|---|
| Scenario 1 | 15 min (1-step) | 23.77% | 10.44% | 9.32% | 7.91% | 15.13% |
| | 30 min (2-step) | | 13.43% | 11.07% | 10.26% | 7.28% |
| | 45 min (3-step) | | 18.37% | 20.06% | 19.55% | 2.53% |
| | 60 min (4-step) | | 21.98% | 24.08% | 23.92% | 0.68% |
| Scenario 2 | 15 min (1-step) | 29.23% | 12.34% | 10.43% | 9.01% | 13.57% |
| | 30 min (2-step) | | 16.74% | 11.89% | 10.93% | 8.09% |
| | 45 min (3-step) | | 21.32% | 18.76% | 18.03% | 3.89% |
| | 60 min (4-step) | | 25.35% | 22.11% | 22.19% | −0.34% |
| Scenario 3 | 15 min (1-step) | 26.47% | 16.17% | 10.72% | 10.37% | 3.22% |
| | 30 min (2-step) | | 22.69% | 13.26% | 12.89% | 2.76% |
| | 45 min (3-step) | | 27.83% | 20.18% | 20.25% | −0.33% |
| | 60 min (4-step) | | 32.47% | 25.84% | 25.88% | −0.17% |

Scenario 3 with a p-value of 0.2114.

In Scenario 1, links with high cross-correlation coefficient and highly consistent geographical attributes are selected to transfer. It can be seen that for both one-step ahead and multi-step ahead prediction, all the results of the transferred model are more accurate than the *baseline network B2*, and the improvement is 15.13% in MAPE value for 1-step ahead prediction. However, as the prediction horizon increases, the improvement gradually decreases, to 7.28% in 2-step ahead prediction, 2.53% in 3-step ahead prediction and 0.68% in 4-step ahead prediction. An example of 1-step ahead prediction with and without the proposed transfer learning method is shown in Fig. 7(a).

Scenario 2 investigates the transfer between links with high cross-correlation coefficients and roughly similar geographical attributes. The transfer results are quite similar to those in Scenario 1. As is shown in Table 4, although the improvement is significant in the 1-step ahead prediction (13.57% MAPE), it gradually decreases to 8.09% in 2-step ahead prediction and 3.89% in 3-step ahead prediction. In the 4-step ahead prediction group, the transfer learning method does not improve the prediction model any more, even negative transfer appears (-0.34% in terms of MAPE). Fig. 7(b) shows an example of time-series plots of 15-minute ahead prediction of the target link M25-4854 with and without transfer learning.

In Scenario 3, links with medium cross-correlation coefficient and highly consistent geographical attributes are tested. In this scenario, the improvement of transfer learning is negligible in all prediction groups (up to 3.22% in 1-step ahead prediction). In addition, although the negative transfer scenario has appeared in 3-step ahead prediction (−0.33%) and 4-step ahead prediction (−0.17%), the difference is so insignificant that we can only infer transfer learning doesn't make any improvements in this scenario. An example of one-step ahead prediction plots of the target link M20-6404 with and without transfer learning is shown in Fig. 7(c).

In summary, when the target link and the source links have consistent data patterns, the proposed hybrid method using DNN and transfer learning can provide more accurate short-term traffic flow prediction for 1-step, 2-step and 3-step ahead prediction. However, the same geographical attributes among links cannot guarantee an improvement in prediction accuracy; even negative transfer might appear if the data patterns between training links and target links are not consistent. As a result, the data patterns are much more important in applying hybrid methods whereas the influence of geographical attributes is small, even negligible.

## 5. Conclusions and future research

This paper has proposed a framework with DNN and transfer learning to improve the transferability of a machine learning based short-term traffic prediction model. A series of experiments have been designed and tested using real-world data collected in the UK. Although these hybrid methods are not investigated in more general scenarios, based on the designed experiments, there is evidence that the hybrid methods with transfer learning and deep learning can improve the transferability of prediction models, hence alleviate the computational burden and enhance prediction accuracy.

In order to investigate factors that affect transferability in link selection criteria, three scenarios with different traffic patterns and geographical attributes have been designed. The results demonstrate that the proposed hybrid framework works when the target link and the selected source links have similar traffic patterns. However, the impacts of geographical attributes are not significant in prediction accuracy. Therefore, in the applied highways dataset, the data patterns are important factors in the application of hybrid transfer methods in the proposed 3-stage framework whereas the influence of geographical attributes is small, even negligible. In Scenarios 1 & 2, where the target link and the source links have similar data patterns, the proposed 3-stage framework can improve machine learning transferability in short-term traffic flow prediction.

We have compared the transfer abilities with different transfer strategies and designs. The results show that: (1) The transferred features and overfitting issues are the two main factors that determine the overall transfer efficacy, either of which may be dominant depending on the transfer strategies and designs; (2) By setting a reasonable initial solution, the transfer learning methods are able to mitigate the overfitting problems, reduce the model training time, guarantee quick convergence and considerably alleviate the computational burden in the traffic prediction applications; (3) By choosing the Transfer Strategy 2 (Fine-tuning with freezing transferred layers), the transferability of machine learning-based prediction models can be improved for one-step ahead prediction, though this improvement gradually decreases in multi-step ahead prediction.

In summary, the main contributions and novelty of this paper include:

(1) We have proposed a novel hybrid deep learning-based architecture with transfer learning techniques, which is able to address insufficient data problems, mitigate model overfitting problems and alleviate the computational burden by improving model's transferability.
(2) The impacts of different input attributes (e.g., geographical attributes and traffic demand) in order to identify selection criteria when transfer learning techniques are used to improve transferability of machine learning tools.
(3) The key components of transfer learning have been analysed and selected through a set of scenarios using real-world traffic data. The recommended parameters and settings of transfer learning can be broadly used with other stand-alone predictors.
(4) We present a straightforward and transferable solution without any data imputation steps for the situation of insufficient training data.
(5) The proposed hybrid prediction model, which is evaluated using real traffic data, can provide reliable one-step ahead results.

Some future research avenues have been opened up based on this paper. In order to investigate impacts of different input attributes and identify selection criteria during transfer, three sources links used in each scenario are selected as typical examples of input attributes. In future applications, more source links should be added into the selection pool and more cases with insufficient training data

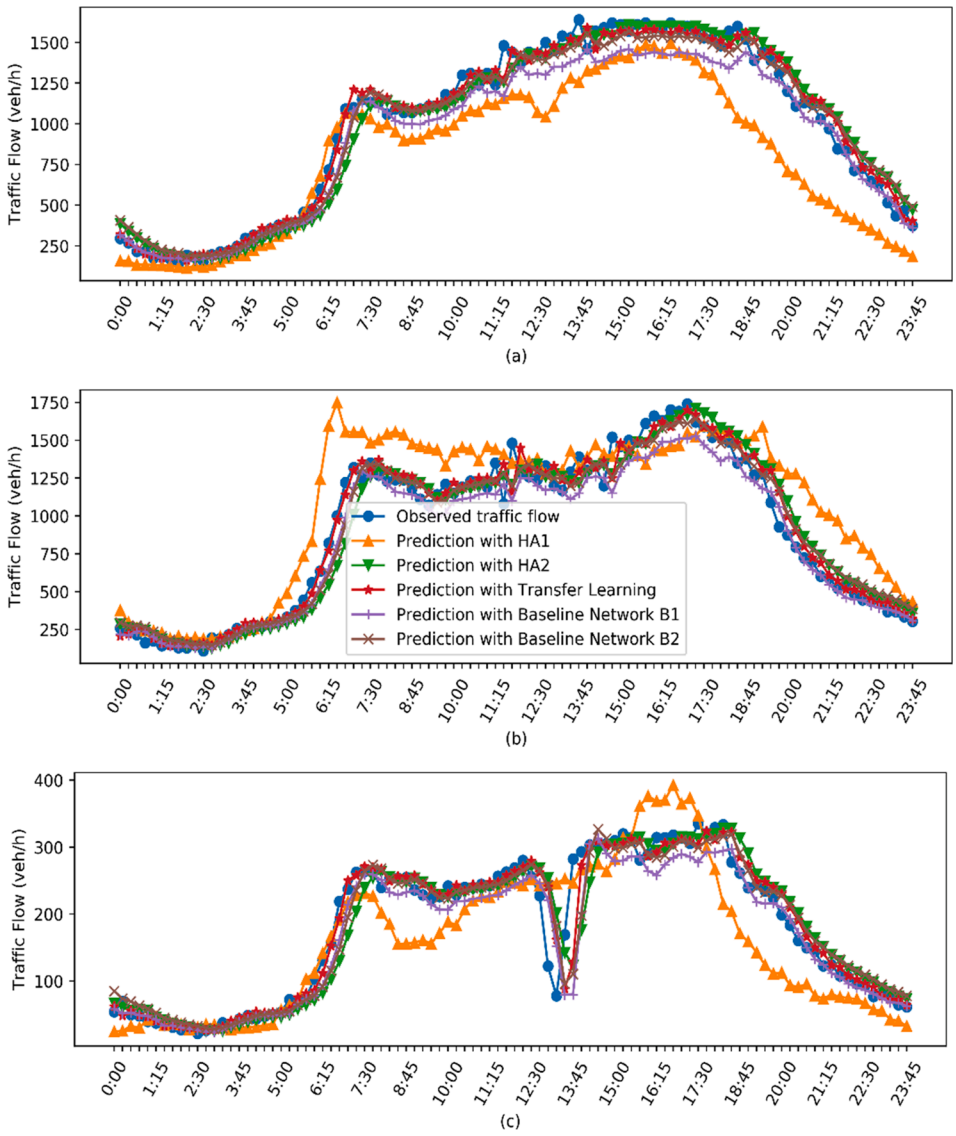



**Fig. 7.** Comparison of predictions with & without transfer learning in three scenarios.

problems should be tested. Moreover, different sub-models for different traffic conditions should be considered via ensemble learning strategies to guarantee more accurate knowledge transfer (Chen et al., 2017b, 2019; Li et al., 2014). Future research on this topic will also look into the transferability of the proposed prediction model framework at a network level and consider geographical attributes like network configurations, sensor distributions, signal settings and the regional points of interest (e.g., shopping malls, governmental organisations), as well as more exogenous variables such as month, weather conditions and incidents, in a theoretically rigorous manner. In addition, although fine-tuning in transfer learning is effective in model transferability improvement, this method is highly dependent on data consistency between source dataset and target dataset. More complicated transfer strategies, such as the deep domain adaptation method, should be investigated in the future. More traffic data collected from arterial roads should be tested.

## CRediT authorship contribution statement

**Junyi Li:** Conceptualization, Methodology, Software, Investigation, Writing - original draft, Visualization. **Fangce Guo:** Conceptualization, Methodology, Supervision. **Aruna Sivakumar:** Supervision. **Yanjie Dong:** Supervision. **Rajesh Krishnan:** Supervision.

## References


Abadi, A., Rajabioun, T., Ioannou, P.A., 2014. Traffic flow prediction for road transportation networks with limited traffic data. IEEE Trans. Intell. Transp. Syst. 16, 653–662.

Badhrudeen, M., Raj, J., Vanajakshi, L.D., 2016. Short-term prediction of traffic parameters performance comparison of a data-driven and less-data-required approaches. J. Adv. Transport. 50, 647–666.

Bekhor, S., Prato, C.G., 2009. Methodological transferability in route choice modeling. Transport. Res. B: Methodol. 43, 422–437.

Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. J. Mach. Learn. Res. 13, 281–305.

Boquet, G., Morell, A., Serrano, J., Vicario, J.L., 2020. A variational autoencoder solution for road traffic forecasting systems: missing data imputation, dimension reduction, model selection and anomaly detection. Transport. Res. C: Emerg. Technol. 115, 102622.

Brand, D., Cheslow, M., 1979. Spatial, temporal, and cultural transferability of travel-choice models. the Fourth International Conference on Behavioral Travel Modeling, 1979 Grainau Bavaria, Germany.

Chen, H., Cui, S., Li, S., 2017a. Application of transfer learning approaches in multimodal wearable human activity recognition. arXiv preprint arXiv:1707.02412.

Chen, H., Grant-Muller, S., Mussone, L., Montgomery, F., 2001. A study of hybrid neural network approaches and the effects of missing data on traffic forecasting. Neural Comput. Appl. 10, 277–286.

Chen, X., Zahiri, M., Zhang, S.C., 2017. Understanding ridesplitting behavior of on-demand ride services: an ensemble learning approach. Transport. Res. C-Emerg. Technol. 76, 51–70.

Chen, X., Zhang, S.C., Li, L., 2019. Multi-model ensemble for short-term traffic flow prediction under normal and abnormal conditions. IET Intel. Transport Syst. 13, 260–268.

Cui, Z., Ke, R., Pu, Z., Wang, Y., 2018. Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. arXiv preprint arXiv:1801.02143.

Cui, Z., Lin, L., Pu, Z., Wang, Y., 2019. Graph Markov network for traffic forecasting with missing data. arXiv preprint arXiv:1912.05457.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T., 2014. Decaf: A deep convolutional activation feature for generic visual recognition. International conference on machine learning, 2014. 647–655.

Du, X., Zhang, H., Van Nguyen, H., Han, Z., 2017. Stacked LSTM deep learning model for traffic prediction in vehicle-to-vehicle communication. 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), 2017. IEEE, 1–5.

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

Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. 12, 2121–2159.

El Esawey, M., Mosa, A.I., Nasr, K., 2015. Estimation of daily bicycle traffic volumes using sparse data. Comput. Environ. Urban Syst. 54, 195–203.

Géron, A., 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, O'Reilly Media.

Guo, F., 2013. Short-term traffic prediction under normal and abnormal conditions. PhD Thesis. Centre for Transport Studies, Imperial College London.

Guo, F., Krishnan, R., Polak, J., 2012. Short-term traffic prediction under normal and incident conditions using singular spectrum analysis and the k-nearest neighbour method. Proceedings of the 17th International Conference on Road Transport Information and Control (RTIC), 2012 London, UK.

Guo, F., Krishnan, R., Polak, J., 2017. The influence of alternative data smoothing prediction techniques on the performance of a two-stage short-term urban travel time prediction framework. J. Intell. Transport. Syst. 21, 214–226.

Guo, F., Krishnan, R., Polak, J. & Luan, J. Short-term car park occupancy prediction in real time. Proceedings of 4th Conference of Transportation Research Group of India (CTRG), 2017b Mumbai, India.

Guo, F., Polak, J.W., Krishnan, R., 2018. Predictor fusion for short-term traffic forecasting. Transport. Res. C: Emerg. Technol. 92, 90–100.

Habtemichael, F.G., Cetin, M., 2016. Short-term traffic flow rate forecasting based on identifying similar traffic patterns. Transport. Res. C: Emerg. Technol. 66, 61–78.

Hadayeghi, A., Shalaby, A.S., Persaud, B.N., Cheung, C., 2006. Temporal transferability and updating of zonal level accident prediction models. Accid. Anal. Prev. 38, 579–589.

Hinton, G., Srivastava, N., Swersky, K., 2012. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent [Online]. Available: http://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf [Accessed 19th July 2019].

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780.

Hu, Q., Zhang, R., Zhou, Y., 2016. Transfer learning for short-term wind speed prediction with Deep Neural Networks. Renew. Energy 85, 83–95.

Huang, J.-T., Li, J., Yu, D., Deng, L., Gong, Y., 2013. In: Cross-language knowledge transfer using multilingual Deep Neural Network with shared hidden layers. IEEE, Vancouver, BC, Canada, pp. 7304–7308.

Huang, W., Song, G., Hong, H., Xie, K., 2014. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans. Intell. Transp. Syst. 15, 2191–2201.

Innamaa, S., 2000. Short-term prediction of traffic situation using MLP-neural networks. 7th World Congress on Intelligent Transportation Systems, 2000 2000 Turin, Italy. 6–9.

Ishak, S., Kotha, P., Alecsandru, C., 2003. Optimization of dynamic neural network performance for short-term traffic prediction. Initiat. Inform. Technol. Geospatial Sci. Transport. 45–56.

Karparthy, A., 2017. A peek at trends in machine learning [Online]. Available: https://medium.com/@karpathy/a-peek-at-trends-in-machine-learning-ab8a1085a106 [Accessed 27th July 2019].

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P., 2016. On large-batch training for deep learning: Generalization gap and sharp minima. arXiv preprint arXiv:1609.04836.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kisgyörgy, L., Rilett, L.R., 2002. Travel time prediction by advanced neural network. Periodica Polytechnica Civil Engineering 46, 15–32.

Kumar, S.V., Vanajakshi, L., 2015. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. Eur. Transp. Res. Rev. 7, 21.

Laharotte, P.-A., Billot, R., El Faouzi, N.-E., Rakha, H. A., 2015. Network-wide traffic state prediction using bluetooth data. Proceedings of the Transportation Research Board 94th Annual Meeting, 2015 Washington D.C, USA.

Lai, G., Chang, W.-C., Yang, Y., Liu, H., 2018. Modeling long-and short-term temporal patterns with Deep Neural Networks. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018 Ann Arbor, MI, USA. ACM, 95–104.

Laña, I., Olabarrieta, I., Vélez, M., Del Ser, J., 2018. On the imputation of missing data for road traffic forecasting: new insights and novel techniques. Transport. Res. C: Emerg. Technol. 90, 18–33.

Li, L., Chen, X.Q., Zhang, L., 2014. Multimodel ensemble for freeway traffic state estimations. IEEE Trans. Intell. Transp. Syst. 15, 1323–1336.

Lippi, M., Bertini, M., Frasconi, P., 2013. Short-term traffic flow forecasting: an experimental comparison of time-series analysis and supervised learning. IEEE Trans. Intell. Transp. Syst. 14, 871–882.

Little, R.J., Rubin, D.B., 1987. Statistical analysis with missing data. John Wiley & Sons Press.

Luan, J., Guo, F., Polak, J. W., Hoose, N., Krishnan, R., 2018. Investigating the transferability of machine learning methods in short-term travel time prediction. Proceedings of the 97th Annual meeting of Transportation Research Board, 2018 Washington DC, USA.

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

Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y., 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transport. Res. C: Emerging Technol. 54, 187–197.

Padiath, A., Vanajakshi, L., Subramanian, S.C., Manda, H., 2009. Prediction of traffic density for congestion analysis under Indian traffic conditions. In: Proceedings of 12th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2009 St. Louis, MO, USA. IEEE, 1-6.
Pamuła, T., 2019. Impact of Data Loss for Prediction of Traffic Flow on an Urban Road Using Neural Networks. IEEE Trans. Intell. Transp. Syst. 20, 1000–1009.
Pan, S.J., Yang, Q., 2009. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22, 1345–1359.
Pascale, A. & Nicoli, M. Adaptive Bayesian network for traffic flow prediction. 2011 IEEE Statistical Signal Processing Workshop (SSP), 2011. IEEE, 177–180.
Polson, N.G., Sokolov, V.O., 2017. Deep learning for short-term traffic flow prediction. Transport. Res. C: Emerg. Technol. 79, 1–17.
Radiuk, P.M., 2017. Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. Inform. Technol. Manage. Sci. 20, 20–24.
Sak, H., Senior, A., Beaufays, F., 2014. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. arXiv preprint arXiv:1402.1128.
Segev, N., Harel, M., Mannor, S., Crammer, K., El-Yaniv, R., 2016. Learn on source, refine on target: a model transfer learning framework with random forests. IEEE Trans. Pattern Anal. Mach. Intell. 39, 1811–1824.
Shao, H., Soong, B.-H., 2016. Traffic flow prediction with long short-term memory networks (LSTMs). 2016 IEEE Region 10 Conference (TENCON), 2016. IEEE, 2986-2989.
Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S., 2014. CNN features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 806–813.
Stanford University, 2019. Transfer Learning and Fine-tuning Convolutional Neural Networks. [Online]. Available: https://cs231n.github.io/transfer-learning/ [Accessed 27th July 2019].
Stathopoulos, A., Karlaftis, M.G., 2001. Spectral and cross-spectral analysis of urban traffic flows. Procedings of the 2001 IEEE Intelligent Transportation Systems (ITSC) (Cat. No. 01TH8585), 2001 Oakland, CA, USA. IEEE, 820-825.
Tian, Y., Zhang, K., Li, J., Lin, X., Yang, B., 2018. LSTM-based traffic flow prediction with missing data. Neurocomputing 318, 297–305.
Tieleman, T., Hinton, G., 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Netw. Mach. Learn. 4, 26–31.
Treiber, M., Helbing, D., 2002. Reconstructing the spatio-temporal traffic dynamics from stationary detector data. Cooperative Transport. Dyn. 1, 3.1-3.24.
Van Lint, J.W.C., 2004. Reliable travel time prediction for freeways. PhD Thesis. Delft University of Technology, Delft, The Netherlands.
Vanderplas, J., 2016. Python data science handbook: Essential tools for working with data. O'Reilly Media Inc.
Vaughan, J., Stoev, S., Michailidis, G., 2013. Network-wide statistical modeling, prediction, and monitoring of computer traffic. Technometrics 55, 79–93.
Vinayakumar, R., Soman, K., Poornachandran, P., 2017. Applying deep learning approaches for network traffic prediction. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017. IEEE, 2353-2358.
Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2014. Short-term traffic forecasting: Where we are and where we're going. Transport. Res. Part C: Emerg. Technol., 43, Part 1, 3–19.
Vu, N.T., Imseng, D., Povey, D., Motlicek, P., Schultz, T., Bourlard, H., 2014. Multilingual Deep Neural Network based acoustic modeling for rapid language adaptation. Proceedings of the 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP), 2014 Florence, Italy. IEEE, 7639-7643.
Wang, J., Chen, R., He, Z., 2019. Traffic speed prediction for urban transportation network: A path based deep learning approach. Transport. Res. C: Emerg. Technol. 100, 372–385.
Wei, W., Wu, H., Ma, H., 2019. An autoencoder and LSTM-based traffic flow prediction method. Sensors 19, 2946.
Xin, X., Liu, Z., Huang, H., 2014. A nonlinear cross-site transfer learning approach for recommender systems. International Conference on Neural Information Processing, 2014. Springer, 495-502.
Yasdi, R., 1999. Prediction of road traffic using a neural network approach. Neural Comput. Appl. 8, 135–142.
Yin, H., Wong, S.C., Xu, J., Wong, C.K., 2002. Urban traffic flow prediction using a fuzzy-neural approach. Transport. Res. C: Emerg. Technol. 10, 85–98.
Yosinski, J., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in Deep Neural Networks? Adv. Neural Inform. Process. Syst. 3320–3328.
Zaremba, W., Sutskever, I., Vinyals, O., 2014. Recurrent neural network regularization. arXiv preprint arXiv:1409.2329.
Zhang, Y., Zhang, Y., 2016. A comparative study of three multivariate short-term freeway traffic flow forecasting methods with missing data. J. Intell. Transport. Syst. 20, 205–218.
Zhang, Z., Li, M., Lin, X., Wang, Y., He, F., 2019. Multistep speed prediction on traffic networks: a deep learning approach considering spatio-temporal dependencies. Transport. Res. Part C: Emerg. Technol. 105, 297–322.
Zhao, Z., Chen, W., Wu, X., Chen, P.C.Y., Liu, J., 2017. LSTM network: a deep learning approach for short-term traffic forecast. IET Intel. Transport Syst. 11, 68–75.
Zhou, L.X., Zhang, S.C., Yu, J.R., Chen, X.Q., 2020. Spatial-temporal deep tensor neural networks for large-scale urban network speed prediction. IEEE Trans. Intell. Transp. Syst. 21, 3718–3729.