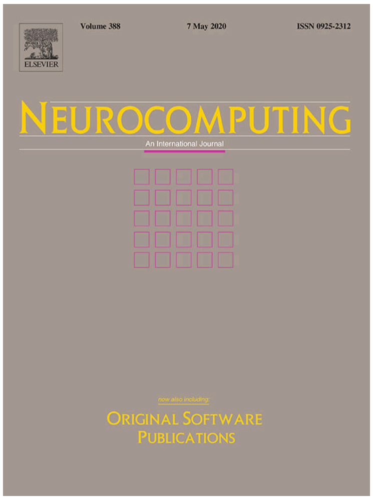


A Joint Temporal-Spatial Ensemble Model for Short-Term Traffic Prediction

Ge Zheng, Wei Koong Chai, Vasilis Katos, Michael Walton

DOI: https://doi.org/10.1016/j.neucom.2021.06.028

To appear in: *Neurocomputing*

Received Date: 21 October 2020
Accepted Date: 12 June 2021

Please cite this article as: G. Zheng, W.K. Chai, V. Katos, M. Walton, A Joint Temporal-Spatial Ensemble Model for Short-Term Traffic Prediction, *Neurocomputing* (2021), doi: https://doi.org/10.1016/j.neucom.2021.06.028

A Joint Temporal-Spatial Ensemble Model for Short-Term Traffic Prediction

Ge Zheng[a], Wei Koong Chai[a], Vasilis Katos[a], Michael Walton[b]

[a]Department of Computing and Informatics, Bournemouth University, Dorset, BH12 5BB, U.K

[b]School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K

Abstract

In this paper, we address the problem of short-term traffic flow prediction since accurate prediction of short-term traffic flow facilitates timely traffic management and rapid response. We advocate deep machine learning approach and propose a novel ensemble model, named ALLSCP, that considers both temporal and spatial characteristics of traffic conditions. Specifically, we consider (1) short-, medium- and long-term temporal traffic evolution, (2) global and local spatial traffic patterns and (3) the correlation of temporal-spatial features in our predictions. We use real-world traffic data from two locations (i.e., Los Angeles and London) with frequent fluctuations (due to proneness to traffic accidents and/or congestion) to train and test our model. For each location, we consider road segments with and without junctions (i.e., linear vs intersection). We compare our model against well-known existing machine/deep learning prediction models. Our results indicate that our ALLSCP model consistently achieves the most accurate predictions ($\approx 96\%$ accuracy both on linear and intersection roadways) when compared against existing models in the literature. In addition, we conducted ablation experiments to further gain insights into the contributions of individual constituent models of our ensemble ALLSCP model. Our results indicate that ALLSCP achieves the best results and is also robust against emergent traffic situations.

Keywords: Short-term traffic prediction, intelligent transportation system, deep learning, ensemble model

1. Introduction

Traffic flow prediction plays an important role in the Intelligent Transportation System (ITS) which aims to provide innovative services to traffic management authorities and road users [1]. Traffic management authorities use accurate traffic flow prediction to optimize advanced traveler information systems, advanced traffic management systems, advanced public transportation systems and commercial vehicle operations [2]. It also facilitates reliable traffic control for alleviating traffic congestion, reducing carbon emissions and improving traffic system efficiency. For road users, it helps them to make better travel decisions (e.g., avoid congested routes) to reduce travel time and cost by offering synchronous traffic information [3][4]. Therefore, traffic flow prediction, as a fundamental support for traffic planning and management, is one of the most important tasks in the area of ITS.

In this paper, we focus on short-term traffic flow prediction problem. The aim is to predict the number of vehicles in a targeted region over a short time interval [2]. Generally, short-term traffic flow prediction mainly depends on historical and real-time traffic data collected from various sensors, e.g., inductive loops, cameras, global positioning system, crowdsourcing, etc. [5]. In early works, traffic flow prediction problem was mainly considered as a pure time series prediction process. For example, [6] developed the autoregressive moving average (ARMA) model for time series analysis, which was a fundamental method taking short-term traffic flow prediction as a temporal process. Based on the ARMA model, [7] built the autoregressive integrated moving-average (ARIMA) model for analyzing freeway traffic time series data. Van Der Voort in [8] further combined a Kohonen map with ARIMA model (named KARIMA) for predicting traffic flow. The subset ARIMA [9] was later developed for short-term traffic flow prediction and claimed to achieve more stable and accurate results compared to other time-series models. Meanwhile, [10] argued that seasonal patterns could be exploited to improve prediction accuracy and proposed the seasonal ARIMA (SARIMA) model.

The above ARIMA-based models assumed a linear relationship between traffic flow data in different time intervals. However, [11] pointed that a non-linear relationship exists between traffic flow data and that purely using ARIMA-based model is inadequate due to their inability to process non-linear relationship. To account for this non-linearity, machine learning algorithms, that have been widely used for solving non-linear prediction problems in various different fields such as in the chemical industry [12][13], biology [14], and modern manufacturing systems [15], were then advocated. Artificial neural networks (ANNs) with non-linear activation functions were used for short-term traffic flow prediction in [16], [17] and [18]. Bidisha [19] built a Bayesian time-series model to estimate the parameters of the SARIMA model. An enhanced K-nearest neighbor (K-NN) algorithm is used in [20] for short-term traffic flow prediction. Xie [21] developed a hybrid model for multi-step ahead traffic flow prediction, which included (1) a spectral analysis technique for extracting periodic trend, (2) the ARIMA model for capturing the mean daily traffic flow variation and (3) the GJR-GARCH model for estimating the volatility. Based on the moving average (MA), ARIMA, exponential smoothing (ES), and neural network (NN) models, an aggregation approach for short-term traffic flow prediction was developed in [22].

With new traffic sensors offering finer granularity measurements and communication technologies able to rapidly disseminate information, researchers started to consider the problem as a temporal and spatial process rather than a pure temporal process. Lv [5] indicated that a transportation system is a highly correlated network. By only considering the time dimension, it neglects the benefit that could be reaped from sharing information among neighboring stations. Along this line, a state space model proposed in [23] suggested to feed data from upstream stations to the model to improve predictions at the downstream stations. Similarly, [24] developed an online learning weighted support vector regression (SVR) model which also uses upstream traffic flow data to contribute to the spatial feature. Considering both temporal and spatial correlations, Lv

[5] developed a deep stacked autoencoder model to avoid shallow models missing detailed information. The stacked autoencoder extracts traffic flow features and then trained in a greedy layer-wise fashion. A softmax layer was utilized for final prediction. Zhao and Chen [25] considered temporal-spatial correlation via a two-dimensional network and developed a model based on long short-term memory (LSTM) network. In [26], a deep learning model that combines a linear model fitted using ι_1 regularization with a sequence of $tanh$ layers is proposed to capture nonlinear temporal-spatial features. These works suggest that it is crucial for neighboring stations to share information in an ITS.

More recently, works combining existing models emerged. The main rationale is that individual model may not be capable of simultaneously extracting all types of features and by combining multiple models, it is possible to capture more patterns in the data. In [27], a hybrid model combining convolutional neural network (CNN) and LSTM models, which work for spatial and temporal feature analysis respectively, is developed and trained by a greedy policy for urban traffic flow prediction. Li [28] built a hybrid model based on ARIMA and SVR to improve predictions while [29] developed an end-to-end deep learning architecture that combined CNN and LSTM to generate a Conv-LSTM module for extracting temporal-spatial traffic flow features and used a Bi-LSTM to extract periodic features of traffic data. Wu [30] built a hybrid DNN-BTF model based on a fully-connected neural networks, recurrent neural networks (RNN) and CNN. RNN and CNN respectively extract the temporal and spatial features and the fully-connected neural network computes the final prediction. Zheng [31] developed an attention-based Conv-LSTM module to extract the spatial and short-term temporal features for short-term traffic prediction. Its attention mechanism is designed to distinguish the difference in importance of traffic flow sequences at different times using different weights. Ma [32] concatenates Multilayer Perception (MLP) Neural Network with ARIMA for network-wide traffic prediction, in which the MLP Neural Network focuses on network-scale co-movement patterns of traffic flows while ARIMA is used to extract other traffic features in the residual time series out of MLP Neural Network. In [33], the authors proposed the combination of ARIMA and Generalized Autoregressive Conditional Heteroskedasticity (ARIMA-GARCH) module for predicting recurring or regular traffic pattern while using a Markov module with state membership degree and a wavelet neural network to predict irregular traffic flow. In [34], the authors advocated the use of data denoising schemes such as Empirical Mode Decomposition (EMD), Ensemble Empirical Mode Decomposition (EEMD) and Wavelet (WL)) to suppress the impact of potential outliers in the data while utilizes LSTM for final predictions. Both [33] and [34] use wavelet algorithms but for different purposes. In [33], it is used for predicting irregular traffic patterns while [34] used it for data denoising. Results from these works highlight the potential of hybrid models in achieving better performance than individual single models.

Summarily, the literature indicates (1) better quality data (in volume and granularity), (2) joint consideration of temporal and spatial processes, and (3) combinations of prediction models all contribute to better short-term traffic flow prediction. Taking these into account, we develop a novel ensemble model, named ALLSCP[1], for short-term traffic flow prediction:

- Our ALLSCP ensemble model is capable of fully exploiting and extracting features of real traffic flow data including short-, medium- and long-term temporal as well as global and local spatial features to compute the final prediction.
- Our ALLSCP is generally applicable to different road network structures. Specifically, we consider traffic flows at road segments with and without junctions (linear vs intersection) at two locations (i.e., Los Angeles and London) using real traffic data.
- The literature (e.g., [36][37]) has found close correlation exists between traffic flow and speed. In our model, we have jointly considered traffic speed and the variances for short-term traffic flow prediction.

[1] Its initial conception and preliminary results are published in [35].

- We conducted ablation experiments based on our ALLSCP model. For this, we built five variants of ALLSCP with each variant having a submodule removed and evaluated their performance for both linear roadways and intersections. The results indicate that our ALLSCP outperforms its five variants.
- We conducted comparative study across several well-known existing machine/deep learning models and other hybrid ones against our model and found that ALLSCP performs better than the rest and also robust in different scenarios.

The rest of this paper is organized as follows. We formulate our traffic flow prediction problem in Section 2. In Section 3, we detail our novel ensemble model, the temporal-spatial input generation, its architecture and constituents submodels. We compare the performance of our model against existing models in Section 4 using real traffic data before concluding our work in Section 5.

2. Problem Definition

Considering a road segment of interest with m observation stations located at different points, each station continuously monitors the traffic flow (i.e., vehicle count) and the average vehicle speed at fixed time interval. Let traffic flow and average traffic speed of the i^{th} station at time interval t be f_t^i and s_t^i respectively. Then the sequences for the traffic flow and speed can be written as $f^i = \{f_1^i, f_2^i, \ldots, f_t^i, \ldots, f_T^i\}$ and $s^i = \{s_1^i, s_2^i, \ldots, s_t^i, \ldots, s_T^i\}$ where $t = 1, 2, 3, \ldots, T$ and $i = 1, 2, 3, \ldots, m$. We consider both traffic flow and speed as input since we found in our datasets a linear relationship between the two (i.e., when traffic volume is high, traffic speed recorded drops) which corroborates the findings in [37] [38].

Generally, traffic flow and speed data are collected from sensors installed on the roadsides with k minutes as a time interval (e.g., 5 minutes in [39] and 15 minutes in [40]). Here, we use $k = 15$ minutes whereby we manually integrate three intervals of traffic flow and speed from [39] into the same time interval as in [40] to ensure the uniformity of experiments.

Assume that station i^* is the selected station. Then, given previous measured traffic flow and speed at i^* and its neighboring stations at the t^{th} time interval, the aim is to predict future traffic flow, $\hat{f}^{i^*}_{t+\Delta}$, at station, i^*, for the $(t+\Delta)^{th}$ time interval where Δ is the prediction horizon which is typically equal to 1, 2, 3, or 4 time intervals [5].

We follow [41] and categorize road segments into two types: one without crossroads (hereafter referred to as "linear") and the other with crossroads (hereafter referred to as "intersection"). For linear roadways, we take the traffic flow and speed data of the chosen station and several consecutive stations before and after that station along the road to predict the traffic flow of the chosen station (i.e., we consider both upstream and downstream traffic data). For intersections, we consider consecutive stations along road segments that meet or cross at the junction. Experiment setup details given in Section 4.1.

3. An Ensemble Prediction Framework

3.1. Temporal-Spatial Input Matrix Generation

Three types of temporal features (short-, medium- and long-term temporal features) and two spatial features (global and local spatial features) are extracted from real world traffic data. In [22, 31, 32, 42], it is indicated that the current traffic flow is not only related to the traffic flow in the several previous time intervals but also related to traffic conditions at the same time in previous days and even weeks. Thus, we define: (1) traffic data in the several previous time intervals as short-term temporal features, (2) traffic data at the same time interval in the previous days as medium-term temporal features and (3) traffic data at the same time interval in previous weeks as long-term temporal features. Correspondingly, three temporal matrices, the time-interval temporal matrix T^i_t, the daily temporal matrix D^i_t and the weekly temporal matrix W^i_t, are generated from original traffic flow data as follows: Eq. (1), Eq. (2), and Eq. (3) respectively:

$$T_t^i = \{f_{t-(k_1-1)}^i, f_{t-(k_1-2)}^i, \ldots, f_{t-2}^i, f_{t-1}^i, f_t^i\} \tag{1}$$

$$\begin{aligned} D_t^i = \{&f^i_{(t+\Delta)-\frac{24\times 60}{k}\times k_2}, f^i_{(t+\Delta)-\frac{24\times 60}{k}\times (k_2-1)}, \ldots, \\ &f^i_{(t+\Delta)-\frac{24\times 60}{k}\times 2}, f^i_{(t+\Delta)-\frac{24\times 60}{k}\times 1}\} \end{aligned} \tag{2}$$

$$\begin{aligned} W_t^i = \{&f^i_{(t+\Delta)-\frac{7\times 24\times 60}{k}\times k_3}, f^i_{(t+\Delta)-\frac{7\times 24\times 60}{k}\times (k_3-1)}, \\ &\ldots, f^i_{(t+\Delta)-\frac{7\times 24\times 60}{k}\times 2}, f^i_{(t+\Delta)-\frac{7\times 24\times 60}{k}\times 1}\} \end{aligned} \tag{3}$$

Traffic measurements obtained from neighboring stations could also be used to improve the prediction accuracy [5]. Thus, we define the difference of traffic data between the targeted station and all its neighboring stations as global spatial features. Furthermore, we define the difference of traffic data between adjacent stations in the neighborhood as local-spatial features. As such, to capture spatio-temporal traffic relationships in the transportation network, we generate Eq. (4) in which m is the number of all stations and n is the number of previous time intervals before the $(t+\Delta)^{th}$. In TS_t, besides f_t^i and s_t^i, we also included the traffic flow and speed changes as additional features. Specifically, we follow [36] and define the traffic flow and speed changes at the i^{th} station between time interval t and $(t-1)$ as $\delta_t^{if} = f_t^i - f_{t-1}^i$ and $\delta_t^{is} = s_t^i - s_{t-1}^i$ respectively. In this matrix, the information of the traffic flow, traffic speed, traffic flow difference and traffic speed difference in rows along the time dimension as temporal features, and the same information in columns along the space dimension corresponding to m neighboring stations is regarded as spatial features.

3.2. The ALLSCP Model

Our novel ensemble model, named ALLSCP, exploits the strengths of several submodels in capturing specific types of temporal and spatial features that

$$TS_t = \begin{bmatrix} f^1_{t-(n-1)} & \cdots & f^1_t & s^1_{t-(n-1)} & \cdots & s^1_t & \delta^{1f}_{t-(n-2)} & \cdots & \delta^{1f}_t \\ \delta^{1s}_{t-(n-2)} & \cdots & \delta^{1s}_t & & & & & & \\ f^2_{t-(n-1)} & \cdots & f^2_t & s^1_{t-(n-1)} & \cdots & s^2_t & \delta^{2f}_{t-(n-2)} & \cdots & \delta^{2f}_t \\ \delta^{2s}_{t-(n-2)} & \cdots & \delta^{2s}_t & & & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & & & & & & \\ f^i_{t-(n-1)} & \cdots & f^i_t & s^i_{t-(n-1)} & \cdots & s^i_t & \delta^{if}_{t-(n-2)} & \cdots & \delta^{if}_t \\ \delta^{is}_{t-(n-2)} & \cdots & \delta^{is}_t & & & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & & & & & & \\ f^m_{t-(n-1)} & \cdots & f^m_t & s^m_{t-(n-1)} & \cdots & s^m_t & \delta^{mf}_{t-(n-2)} & \cdots & \delta^{mf}_t \\ \delta^{ms}_{t-(n-2)} & \cdots & \delta^{ms}_t & & & & & & \end{bmatrix} \quad (4)$$

contribute to the final prediction. Fig. 1 shows the architecture of ALLSCP. Matrices T^i_t, D^i_t, W^i_t and TS_t are utilized for short-, medium-, long-term temporal as well as global and local spatial feature extraction by specific submodels as follows:

3.2.1. Short-term Temporal Feature Extraction

In our model, T^i_t is used to extract short-term temporal features. We exploit ARIMA for this purpose. The key idea is that while ARIMA does not consider the non-linearity of traffic data and inherently assumes that traffic data to be linearly correlated with time, changes of traffic status over very short duration are continuous and can be considered as linear [28, 32, 33]. The non-linearity of traffic data is most evident at longer term period when the traffic conditions can change significantly. Therefore, the non-linear patterns will be analysed by other modules of the proposed model from traffic data at medium- and long-

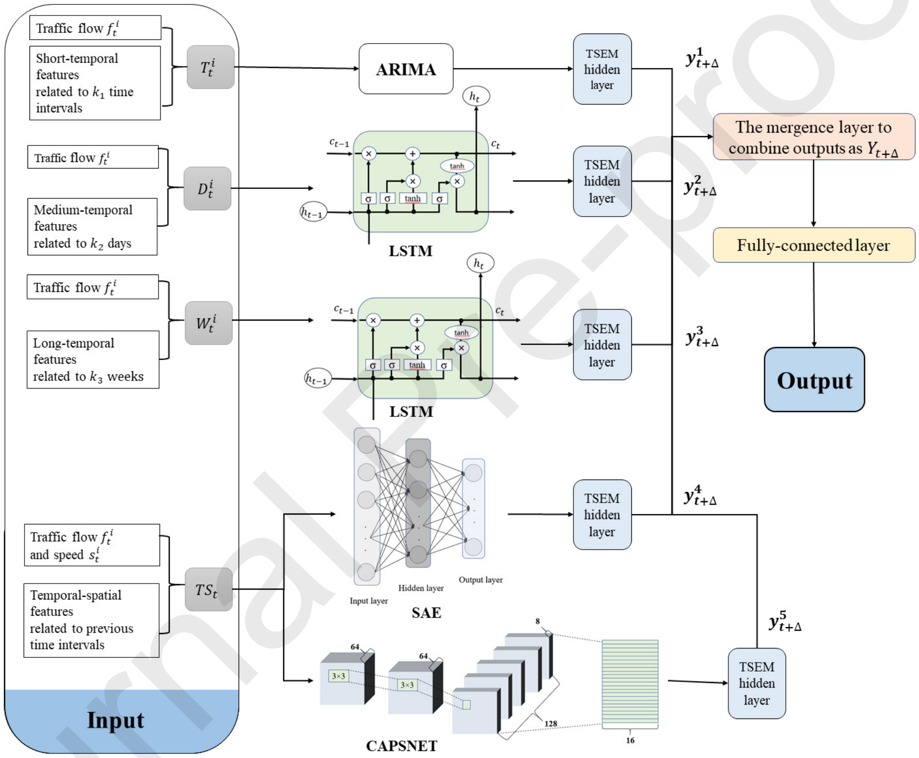


Figure 1: The architecture of the ALLSCP model.

term duration. As such, we use ARIMA model to analyze short-term temporal features and for longer temporal features, we propose to use LSTM (cf. Section 3.2.2). ARIMA is widely used as a statistical model for the prediction of time series data. It requires time series data to be stationary or stationary after differentiating [43]. Traffic flow is a periodic time series and usually has slight changes in a short time interval. Thus, traffic flow is considered as stationary time series and this is also why many existing works used ARIMA for short-term traffic flow prediction (e.g., [7], [8], [9] and [10]). There are three important parameters: (1) p – the number of autoregressive terms, (2) d – the number of non-seasonal differences for converting data to be stationary and (3) q – the number of lagged forecast errors in the prediction equation. Firstly, we ensure our input data fulfill the stationary property by differentiating the input d times. Then short-term temporal features are extracted from previous p time intervals with the number of lagged prediction errors q. Then short-term temporal features are extracted from the matrix T_t^i in the ARIMA submodel using Eq. (5).

$$\begin{split} f_{t+\Delta}^i = c + \phi_1 f_t^i + \phi_2 f_{t-1}^i + \cdots + \\ \phi_{p-1} f_{t-(k_1-2)}^i + \phi_p f_{t-(k_1-1)}^i \end{split} \tag{5}$$

where ϕ_p is the parameter of the autoregressive part of ARIMA, and c is a constant.

As a part of input for final prediction, the output from ARIMA is formatted to be in the same dimension with other features via the ensuing TSEM hidden layer (see Fig. 1).

As discussed in Section 1, ARIMA does not capture non-linear relationship. For this, we use the following submodels to capture the non-linear relationship in traffic flow.

3.2.2. Medium- and Long-term Temporal Feature Extraction

We use LSTM, which is capable of learning long-term relationship from historical data [44], to extract medium- (daily) and long-term temporal (weekly) features from D_t^i and W_t^i, respectively. LSTM is an extension of the RNN model [45]. Compared to RNN that has one part (i.e., the *tanh* layer) LSTM consists of four parts: three gates (namely, input gate I_t, output gate O_t and forget gate F_t) and a cell state (C_t).

Taking D_t^i as an example on how the LSTM submodel extracts medium-temporal features in our framework, the forget gate F_t with a sigmoid layer, σ_g, firstly determines the part of information in current traffic flow f_t^i and in the last hidden state, H_{t-1} that it needs to forget and update it to the cell state, C_t, via Eq. (6). To supplement the forgotten information by forget gate F_t, the input gate, I_t, with a sigmoid layer σ_g is used to decide the information from current traffic flow f_t^i to be added into the cell state C_t via Eq. (7). Then, the cell state, C_t, is updated using the tangent layer σ_C (cf. Eq. (9),) for integrating the traffic flow information provided from the forget gate, the input gate and the last cell state C_{t-1}. Meanwhile, the output gate with a sigmoid layer σ_g selects previous information remembered by H_{t-1} and the current information f_t^i by Eq. (8) for contributing to final output. Finally, the predicted result is computed by combining remembered information from the output gate O_t and the cell state C_t with a tangent layer σ_H by Eq. (10).

$$F_t = \sigma_g(W_F \times f_t^i + U_F \times H_{t-1} + b_F) \tag{6}$$

$$I_t = \sigma_g(W_I \times f_t^i + U_I \times H_{t-1} + b_I) \tag{7}$$

$$O_t = \sigma_g(W_O \times f_t^i + U_O \times H_{t-1} + b_O) \tag{8}$$

$$C_t = F_t \circ C_{t-1} + I_t \circ \sigma_C(W_C \times f_t^i + U_C \times H_{t-1} + b_C) \tag{9}$$

$$H_t = O_t \circ \sigma_H \times (C_t). \tag{10}$$

where W_F, W_I, W_O and W_C are the weights of the forget gate, the input gate, the output gate and the cell state respectively while b_F, b_I, b_O and b_C are the corresponding bias for each gate and state. Furthermore, U_F, U_I, U_O and U_C

are the weights of the last hidden state H_{t-1}. We use σ_g to denote a sigmoid function ($= \frac{1}{1+e^{-x}}$) in three gates and the operator $\circ$ to denote Hadamard product. σ_C and σ_H are hyperbolic tangent functions ($tanh(x)$) for the cell state and the final output. In our case, one LSTM submodel, for extracting medium-temporal features from D^i_t, has k_2 memory units and another LSTM submodel, for extracting long-temporal features from W^i_t, has k_3 memory units. This means we use traffic flow in k_2 time intervals in previous days and in k_3 time intervals in previous weeks to extract medium- and long-temporal features.

3.2.3. Global-Spatial Feature Extraction

The input matrix TS_t that records historical traffic flow in the targeted station and their neighboring stations is used to extract global spatial features by utilizing the stacked autoencoder (SAE) in our framework in Fig. 1. The SAE neural network [46], as an unsupervised learning algorithm, can learn features from high dimension data and then encode it into low dimension data. If a predictor (e.g., a logistic regression layer [47] or a softmax layer [48]) is added on the top of SAE model, it can be used for prediction problems. The SAE submodel in our framework includes one input layer and n_s hidden layers for global spatial feature extraction. It first takes the input matrix TS_t into the SAE submodel via the input layer. Then, it encodes the output of the input layer to the 1^{st} hidden layer representation $y^1(TS_t)$ via Eq. (11) and finally it decodes the representation $y^1(TS_t)$ back into a reconstruction $z^1(TS_t)$ calculated via Eq. (12). The shape of the input matrix TS_t is $(4n-2) \times m$ so the input layer has $(4n-2) \times m$ neural units without any weighted inputs. The number of neural units in the hidden layers is set as n_u that is decided by the grid search detailed in [49] from a limited range. We consider logistic sigmoid function for each hidden layer for extracting global spatial features from the input matrix TS_t by the fully connection between layers.

$$y^1(TS_t) = G\Big(w_1 \times TS_t + b_1\Big) \tag{11}$$

$$z^1(TS_t) = Z\Big(w_2 \times y(TS_t) + b_2\Big) \tag{12}$$

where G is the logistic sigmoid function as an encoder, w_1 is the weight vector of the encoder, and b_1 is the bias vector. Correspondingly, Z is the logistic sigmoid function as a decoder, and w_2 and b_2 are respectively the weight vector and the related bias vector of the decoder.

3.2.4. Local-Spatial Feature Extraction

We extract local-spatial features via CAPSNET submodel. Compared to SAE that extracts global-spatial features via full connection between layers, CAPSNET focuses on local-spatial features via the local connections implemented by the kernel functions in convolutional layers. For example, if the kernel size was set as 3 × 3, convolutional value only includes features of three continuous points inside the kernel. This can be considered as the local feature. Capsule network [50] is based on CNN. CAPSNET is characterized by "capsules" in vector form. When extracting local features in images, important local information that the capsules detect is encapsulated in a vector form. The length of an output vector encodes the probability of a feature and the direction of the vector encodes the gesture of features, such as rotation angle and direction. Compared to CNN, CAPSNET can effectively extract more detailed local-spatial features because of the vector form. Here, the input matrix TS_t is regarded as a image matrix in our CAPSNET submodel, and its shape is $(4n-2) \times m$.

Our CAPSNET submodel consists of two convolutional layers and a fully connected layer, TrafficCaps. The first convolutional layer is same as the convolutional layer in conventional CNN, which is used to extract spatial features of traffic flow between neighboring stations. The *ReLU* function (cf. Eq. (13)) is used as the activation function in this layer. The second convolutional layer is the primary capsule layer to capture the local-spatial features and used for converting the single scalar output of the first convolutional layer into vector form with a dimension of 8 by "capsules". Finally, the TrafficCaps layer extracts the spatial relationship between the local-spatial features obtained from primary capsules and outputs the features to a set of advanced capsules with a dimension of 16. Eq. (14) is the novel nonlinear "squashing" activation function for the vector form of capsules used in the primary convolutional and TrafficCaps layers.

$$ReLU(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases} \tag{13}$$

$$v_j = \frac{||s_j||^2}{1 + ||s_j||^2} \frac{s_j}{||s_j||} \tag{14}$$

where v_j and s_j are the output and input vector of capsule j respectively. The final output of the submodel is a vector.

3.2.5. Final Prediction

After extracting short-, medium-, long-term temporal as well as global and local spatial features, we use a fully-connected layer as a predictor in our framework for the final short-term traffic flow prediction. In addition, on the top of the fully-connected layer, there are two hidden layers (namely TSEM hidden layer and Merging layer). The function of the TSEM hidden layer is to convert outputs from five submodels into the related tensors of the same dimension: $y^1_{t+\Delta}$ from the ARIMA submodel for short-term temporal features, $y^2_{t+\Delta}$ and $y^3_{t+\Delta}$ from the LSTM submodel for medium- (daily) and long-term (weekly) temporal features, $y^4_{t+\Delta}$ from the SAE submodel for global-spatial features and $y^5_{t+\Delta}$ from the CAPSNET submodel for local-spatial features. Generally, there are main three approaches to fuse different features in an ensemble learning model - (1) concatenation ensemble, (2) average ensemble and (3) weighted ensemble. In our work, the five outputs ($y^1_{t+\Delta}$, $y^2_{t+\Delta}$, $y^3_{t+\Delta}$, $y^4_{t+\Delta}$, $y^5_{t+\Delta}$) are concatenated in the last dimension to generate temporal and spatial-fused features $Y_{t+\Delta}$ as Eq. (15) in the mergence layer. Then a fully-connected layer is used to compute the output as the final prediction. Our design here then adopts the weighted ensemble approach as the weights in the fully-connected layer are used to determine the contribution of each type of features (or each modules) to the final prediction.

$$Y_{t+\Delta} = \{y^1_{t+\Delta}, y^2_{t+\Delta}, y^3_{t+\Delta}, y^4_{t+\Delta}, y^5_{t+\Delta}\}. \tag{15}$$

The loss function employed in our model is the Mean Squared Error (MSE) and the optimizer *Adam* [51] is utilized to minimize the loss function MSE. The grid search method [49] is used to find the optimal parameter combinations in the limited range detailed in Section 4.2. In addition, to follow the convention, we use 70% of each dataset for training, 20% for validation and 10% for testing.

4. Experiments and Results

4.1. Data and Experiment Scenario

In our experiments, we use traffic data from two locations: Los Angeles, USA and London, United Kingdom. For each location, we use data on two types of roadways (i.e., linear and intersection road segments). The Los Angeles traffic data is collected from the California Department of Transportation (Caltrans) (PeMS) [39]. PeMS aggregates traffic data into 5-min interval for every station. Meanwhile, the London traffic data is collected from the Highways England [40] where the traffic data is aggregated into 15-min interval each for every station. For both, we use data period between January and June 2018. As prior mentioned, we use 15-min interval for both locations to ensure the uniformity of experiments. We choose roadways prone to heavy congestion and frequent incidents.

For linear roadway in Los Angeles, traffic data is collected from roadway 605. Specifically, data from five observation stations are used to predict the traffic flow at the third observation station (i.e., the middle of the five stations). Fig. 2 shows the five observation stations labeled as a_l, b_l, c_l, d_l and e_l, and we predict short-term traffic flow at station c_l. Thus, the number of observation stations

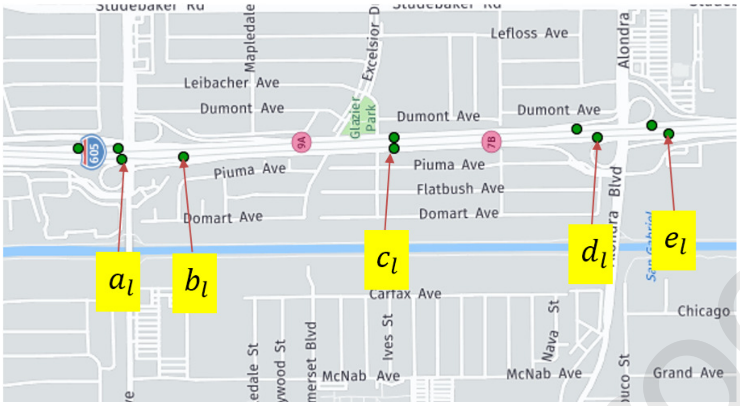


Figure 2: Five observation stations for the chosen road in Los Angeles. We predict the traffic flow at station c_l.

inside of temporal-spatial matrix TS_t, $m = 5$. The two observation stations before and after station c_l are taken into consideration for spatial features.

For intersections in Los Angeles, we focus on the junction between road 605 and 105. Eight contiguous observation stations are utilized to predict two targeted stations. Fig. 3 shows the eight selected observation stations (labeled as a_i, b_i, c_i, d_i, e_i, f_i, g_i, h_i). We consider the traffic flow in the direction from a_i moving towards d_i, f_i and h_i. Our aim is to predict short-term traffic flow at e_i and g_i. The number of observation stations inside of temporal-spatial matrix TS_t, $m = 8$. Compared to the previous case for linear roadway, the difference is that the data at the exit and entrance corresponding to the current direction is considered into prediction. This is important as drivers may avoid congested road segments by exiting at the junction and return back further down the road. For example, in Fig. 3, if b_i and/or g_i is congested, drivers approaching h_i from a_i can choose to exit to c_i or e_i rather than going directly from a_i to h_i. Correspondingly, similar linear and intersection scenarios are considered for London. Traffic data on linear roadways in London is collected on the M4 motorway (cf. Fig. 4), and for intersection, we use the data at the junction between M4 and M25 (cf. Fig. 5).

Hereinafter, we label our datasets as follows: (1) `L-Los` – linear road in

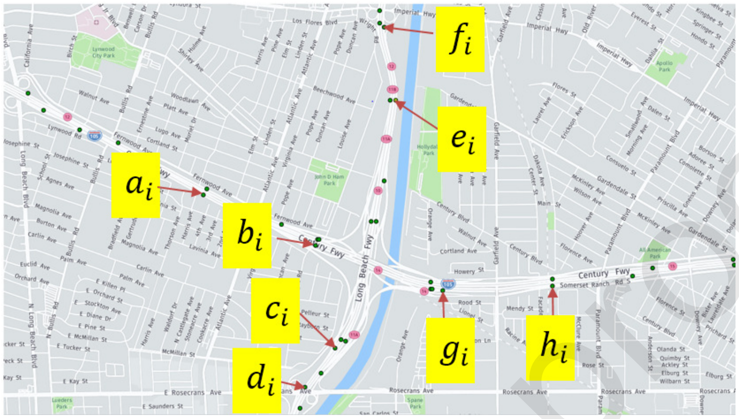


Figure 3: Eight observation stations for the chosen road in Los Angeles. We predict the traffic flow at station: e_i and g_i.

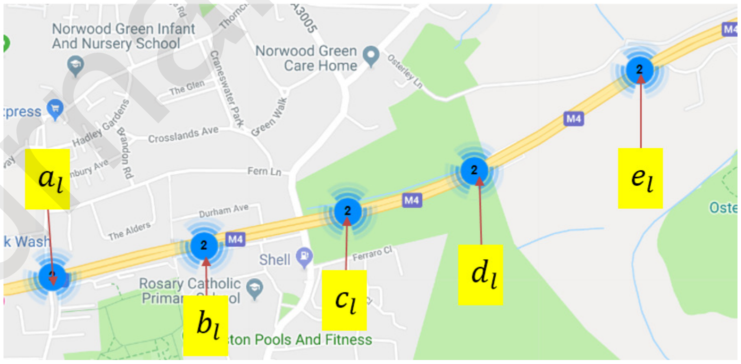


Figure 4: Five observation stations for the chosen road in London. We predict the traffic flow at station c_l.

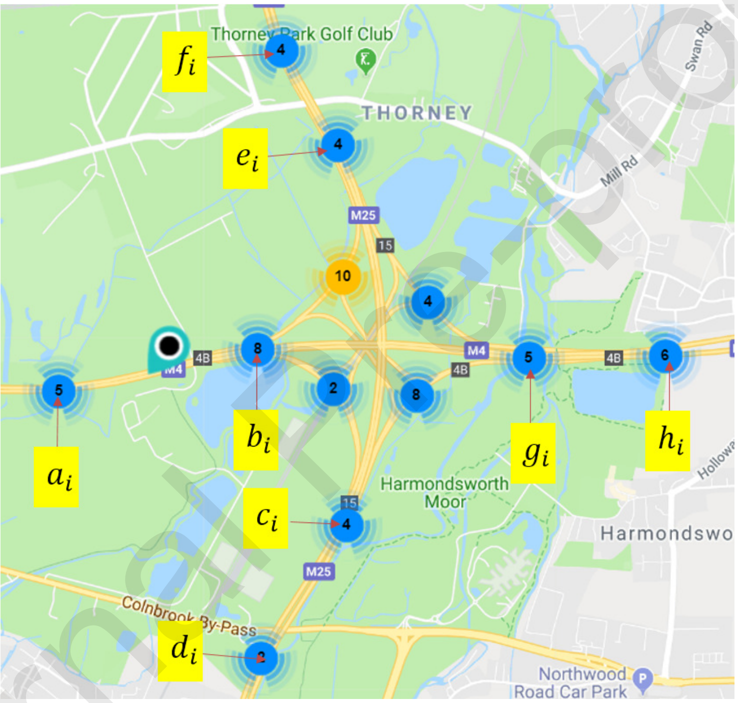


Figure 5: Eight observation stations for the chosen road in London. We predict the traffic flow at stations: e_i and g_i.

Los Angeles, (2) L-London – linear road in London, (3) I-Los – intersection road in Los Angeles, (4) I-London – intersection road in London. Besides using the original traffic flow data, we also test our model against pre-processed data generated using Hodrick Prescott (HP) filter [52]. HP filter is a data-smoothing techniques to remove short-term fluctuations and reveal long-term trend. Pre-processed (or de-noised) datasets corresponding to original datasets are named as PL-Los, PL-London, PI-Los and PI-London.

4.2. Model Setting

The ALLSCP parameters that we tune are as follows:

1. ***Parameters in the temporal-spatial input matrix:*** By analyzing the autocorrelation function (ACF) and partial autocorrelation function (PACF) of traffic flow sequences, we set k_1 in T_t^i to 9 as we found traffic flow in the next time interval highly depends on traffic flow in 9 previous time intervals. Both k_2 in the daily temporal matrix D_t^i and k_3 in the weekly temporal matrix W_t^i are set to 7, taking into account 7 previous days and weeks respectively. The number of previous time intervals n and the number of stations m in the TS_t are respectively set as 3 and 5 in the linear roadways and 3 and 8 in the intersections, (See Section 4.1).

2. ***Parameters in the ARIMA submodel:*** Lag order, p, differentiating times, d, and moving average window, q are respectively set to 9 (equal to k_1), 1 and 0, which are decided by analyzing the autocorrelation coefficient and partial autocorrelation coefficient of matrix T_t^i.

3. ***Parameters in the LSTM submodel:*** The parameters needed to tune in our LSTM submodels are memory units n_d for the input matrix D_t^i and n_w for the input matrix W_t^i. Based on the number of time intervals defined in the input matrix D_t^i and W_t^i, we set $n_d = k_2$ and $n_w = k_3$.

4. ***Parameters in the SAE submodel:*** For SAE, we search the optimal number of hidden layers n_s between 1 and 6, and the number of neurons

n_{su} in each of hidden layers from 200 to 400 (step size = 50). More layers can capture more information from input, but it costs more time.

5. ***Parameters in the CAPSNET submodel:*** The parameters for CAPSNET is given in Table 1. There are four layers including two conventional convolutional layers, one primary capsule layer (namely PrimaryCaps) and one traffic capsule layer (namely TrafficCaps). The two conventional convolutional layers are used to capture the temporal-spatial features of short-term traffic flow, in which the kernel size in both conventional convolutional layers and activation function are respectively 3×3 and "ReLU". Convolution operations are performed with 2 as the stride and zero padding. The PrimaryCaps layer is a convolutional layer with 128 channels with 3×3 kernel size. It has 16 (128/8) capsules and each capsule is an 8-dimensional vector. The difference when compared to conventional convolutional layers is that the activation function in this layer is "Squashing" function (cf. Eq. (14)) rather than "ReLU". This activation function is also used in the TrafficCaps layer with 16 advanced capsules and each of capsules has a 16-dimensional vector. The advantage of using this in our work is that it produces output in a vector consisting of 16 values to allow us taking more traffic information than a scalar value obtained by other activation functions.

Table 1: Parameter Setting in the CAPSNET submodel

Layer name	Parameter	Activation
Convolution	(3, 3, 64)	ReLU
PrimaryCaps	(3, 3, 128) Capsule dimension = 8	Squashing
TrafficCaps (Fully connected)	Advanced capsule = 16 Capsule dimension = 16	Squashing
(Flattened)	256	

4.3. Performance Metrics

For evaluation, we follow [5][53] and define prediction accuracy of short-term traffic flow prediction as $(1 - MRE)\%$ where Mean Relative Error (MRE) is given as:

$$MRE = \frac{1}{N} \cdot \sum_{t=1}^{N} \frac{|f_t^i - \hat{f}_t^i|}{f_t^i} \tag{16}$$

MRE is the relative difference between the predicted and real traffic flow and is utilized to measure relative prediction error. Furthermore, we complement this with two other conventional performance metrics commonly used in the literature [22][5], namely Mean Absolute Error (MAE) and Root-Mean Square Error (RMSE) which are computed as follows:

$$MAE = \frac{1}{N} \cdot \sum_{t=1}^{N} |f_t^i - \hat{f}_t^i| \tag{17}$$

$$RMSE = \left[\frac{1}{N} \cdot \sum_{t=1}^{N} (f_t^i - \hat{f}_t^i)^2 \right]^{\frac{1}{2}} \tag{18}$$

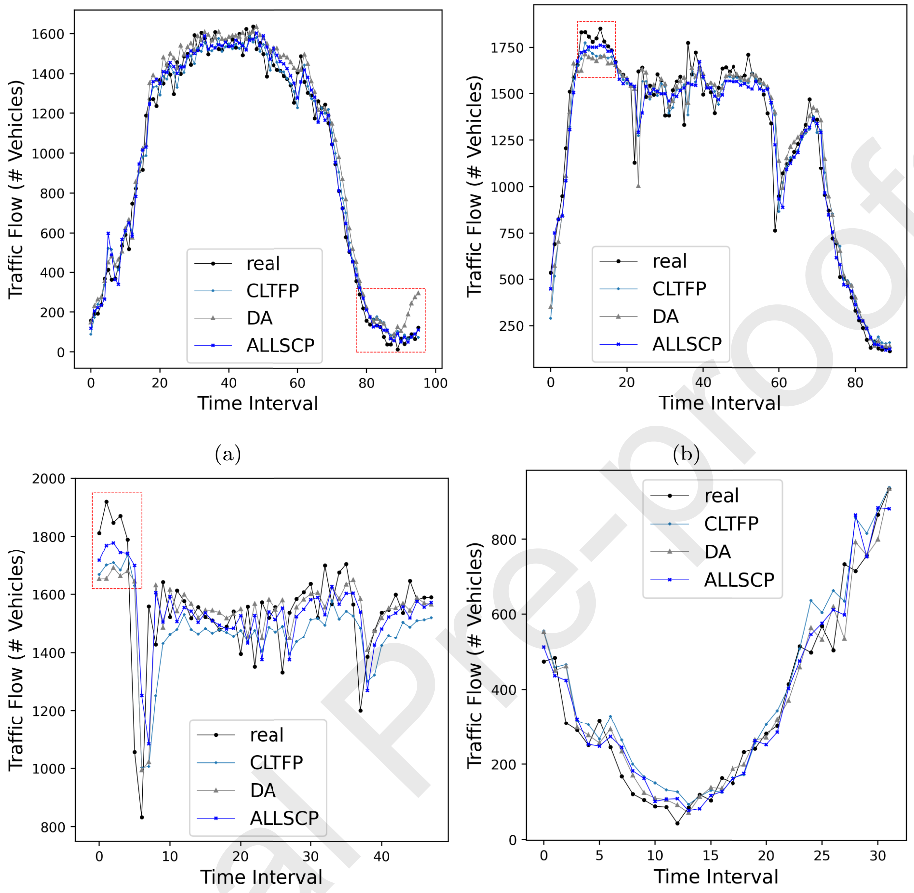
MAE presents the average absolute difference between the predicted traffic flow and real traffic flow. It is used to measure absolute prediction error. RMSE is the standard deviation of the residuals where residual is the difference between predicted traffic flow and real traffic flow.

4.4. Results and Discussion

We compare the performance of our ALLSCP against a time-series prediction model (ARIMA), a simple machine learning model (SVR), four deep learning models (LSTM, SAE, CNN and CAPSNET) and two ensemble models (DA [22] and CLTFP [42]) on linear and intersection roadways.

4.4.1. Linear Roadways

We compare the original traffic flow against the predicted traffic flow from our ALLSCP model and two other existing ensemble models (i.e., CLTFP ad



(a)

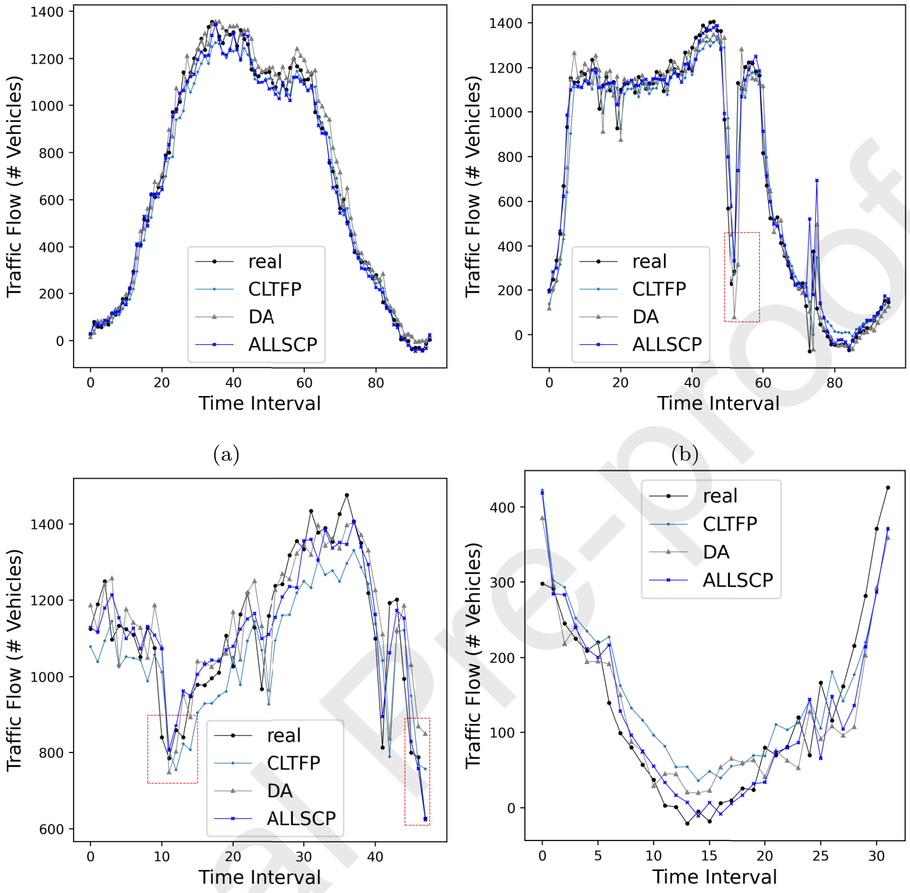
(b)

(c)

(d)

Figure 6: (Color Online) Real (black) and predicted traffic flow by our ALLSCP (blue) and other two existing ensemble models (other colors) in different traffic situations: (a) Normal traffic condition (24-hour period), (b) Abnormal traffic condition (24-hour period), (c) Peak hours (12-hour period), (d) Off peak hours (8-hour period), for `L-Los`. Red boxes in sub-figures highlight obvious differences between ALLSCP and the other two existing models.

DA) for `L-Los` in Fig. 6 and `L-London` in Fig. 7 under different traffic situations. Fig. 6(a) and Fig. 7(a) present the traffic flow over a 24-hour period with normal traffic condition from `L-Los` and `L-London` respectively, while Fig. 6(b) and Fig. 7(b) show the traffic flow over a 24-hour period when the traffic condition is affected with various traffic incidents. In addition, Fig. 6(c) and Fig. 7(c) show the traffic flow during rush hours over a period of 12 hours, and finally,



(a) (b)

(c) (d)

Figure 7: (Color Online) Real (black) and predicted traffic flow by our ALLSCP (blue) and other two existing ensemble (other colors) models in different traffic situations: (a) Normal traffic condition (24-hour period), (b) Abnormal traffic condition (24-hour period), (c) Peak hours (12-hour period), (d) Off peak hours (8-hour period), for `L-London`. Red boxes in sub-figures show obvious differences between ALLSCP and the other two existing models.

Fig. 6(d) and Fig. 7(d) show the off-peak time over a period of 8 hours. In all cases, especially for Fig. 6(c) and Fig. 7(c) during rush hours, our model can both capture sudden changes as well as finer traffic changes. To further show these, we added red boxes in all sub-figures to highlight the differences between ALLSCP and the other two existing ensemble models. Overall, ALLSCP captures the traffic flow changes, following closely the diurnal pattern exhibited in road traffic, even under abnormal traffic conditions.

Fig. 8 shows the accuracy (i.e., $(1 - MRE)\%$) of the models across `L-Los`, `PL-Los`, `L-London` and `PL-London` datasets. For all cases, our ALLSCP achieves the best accuracy. Specifically for cases using original traffic data, ALLSCP achieves accuracy of 93.86% and 95.05% for Los Angeles and London respectively while the rest of the models on average only achieve accuracy of 91.84% and 92.59%. Amongst the considered models, DA and CAPSNET are the second best models for `L-Los` and `L-London` respectively with a performance gap of 1.11% and 1.78% when compared to ALLSCP. On the other end of the spectrum, the worst performing model for both cases are CNN as it is only capable of capturing local-spatial features. Furthermore, when we use de-noised data, ALLSCP's accuracy further improved to 98.16% (4.3% improvement) and 97.50% (2.45% improvement) for `PL-Los` and `PL-London` respectively. DA remains to be the closest rival for prediction on Los Angeles traffic with 96.88% accuracy. However, for London, CNN's prediction accuracy improves significantly to become the second best (95.83%). LSTM which mainly focuses on temporal feature extraction is the worst performer when using de-noised London data, indicating the traffic pattern near Heathrow airport is more complex. In fact, we note that while for Los Angeles, all models achieve improved accuracy, this is not the case for London when the prediction accuracy for ARIMA, LSTM, SAE and SVR worsened, again mainly due to the higher volatility in traffic near Heathrow airport. Our ALLSCP model achieves overall higher prediction accuracy due to its ability to capture different temporal (short, medium and long) and spatial (global and local) features. For instance, we exploit CAPSNET's feature on encapsulating important information related to local features into a vector form that can carry more information than a scalar value. From our results, this encapsulation alone is not sufficient but using our ensemble model, we achieve better accuracy. Moreover, due to the periodic properties of traffic data, the carefully designed input (including short, medium and long temporal traffic data on the targeted station and on its neighboring stations) also contributes to the enhancement of the prediction accuracy.

Table 2: Comparison of all models for both line roadways

Model	L-Los			PL-Los			L-London			PL-London		
	MAE	MRE	RMSE	MAE	MRE	RMSE	MAE	MRE	RMSE	MAE	MRE	RMSE
ARIMA	81.89	0.0884	109.55	72.21	0.0662	97.62	59.37	0.0890	92.10	58.64	0.0743	92.32
LSTM	74.56	0.0778	97.42	79.42	0.0696	104.36	79.20	0.1027	106.03	50.25	0.0678	83.49
SAE	59.68	0.0829	113.42	62.35	0.0636	100.87	60.58	0.0815	104.93	53.83	0.0756	111.90
CAPSNET	85.74	0.0781	117.17	36.85	0.0329	48.96	44.69	0.0673	80.18	31.39	0.0436	53.14
CNN	95.91	0.0885	128.60	42.94	0.0403	68.00	58.58	0.0866	105.15	32.92	0.0417	99.62
SVR	76.39	0.0803	103.53	57.74	0.0559	81.95	51.70	0.0768	82.08	56.83	0.0728	95.58
CLTFP	**40.67**	0.0845	**50.46**	**13.28**	0.0378	**19.67**	**30.19**	0.0783	**40.62**	45.48	0.0763	88.52
DA	77.65	0.0725	103.51	14.32	0.0312	22.37	58.56	0.0703	92.21	28.45	0.0521	43.51
ALLSCP	71.64	**0.0614**	95.08	20.10	**0.0184**	26.84	36.29	**0.0495**	65.80	**18.22**	**0.0250**	**25.90**

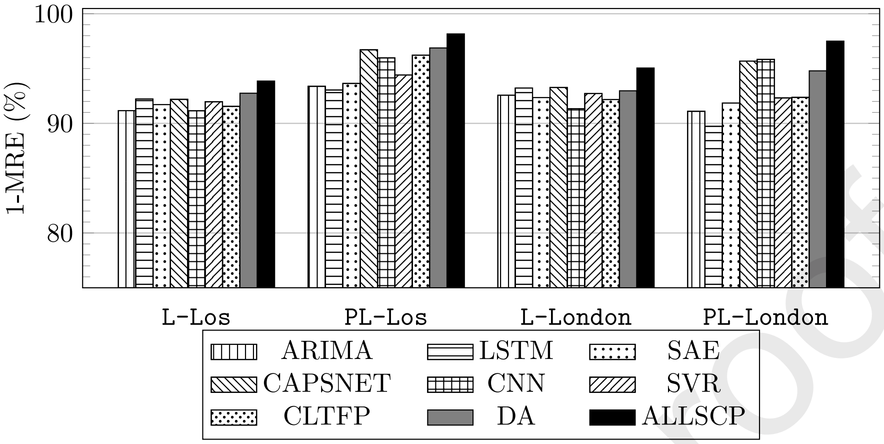


Figure 8: $(1 - MRE)\%$ achieved on `L-Los` (left), `PL-Los` (middle-left), `L-London` (middle-right) and `PL-London` (right) collected from linear roadways. ALLSCP achieves the best accuracy for all cases.

Table 2 presents the MAE, MRE and RMSE achieved by all models on `L-Los`, `PL-Los`, `L-London` and `PL-London`. Between a statistic model and a simple machine learning model (i.e., ARIMA and SVR), SVR performs better. This is mainly due to the non-linear relationship between traffic flow in different time intervals which ARIMA fails to take into account whereas SVR with a non-linear kernel function (i.e., RBF kernel function) is capable of mapping a non-linear vector to a high dimensional feature space for conducting linear regression. Meanwhile, the four deep learning models (i.e., LSTM, SAE, CNN and CAPSNET) generally achieve better predictions compared to simple machine learning models. For instance, LSTM focusing on time-series data using the cell state to store information on long-term dependencies of traffic data outperforms both ARIMA and SVR. For the SAE model, full connection is used between hidden layers. Therefore, it missed the contribution of local features for traffic prediction. Compared to SAE model, CNN can capture local-spatial feature for obtaining better result because of convolutional kernels. Based on CNN, CAPSNET converts scalar values representing features into a vector form to obtain more detailed information for traffic prediction. This is the reason that CAPSNET model obtains best results, especially on PL-Los (96.71%) and PL-London (95.64%). This implies the importance of spatial information for traffic prediction. Furthermore, between the two ensemble models (CLTFP and DA), MRE of DA model is lower on four datasets, and the other two metrics (MAE and RMSE) of CLTFP model are lower except on PL-London. DA model mainly depends on temporal feature extraction for prediction while CLTFP model extracted temporal-spatial features for the final prediction. This again indicates the importance of simultaneously taking the temporal and spatial features on the problem of traffic prediction. Finally, while our ALLSCP consistently achieves the best accuracy, CLTFP relegates it to second best in some cases for MAE and RMSE.

4.4.2. Intersections

For intersections, ALLSCP again achieves the best accuracy across both original and pre-processed datasets at different locations (i.e., at e_i and g_i). We present the $(1 - MRE)\%$ results for e_i in Fig. 9[2]. ALLSCP achieves an average of 95.53% accuracy for both e_i and g_i across all datasets while the average achieved accuracy by the other eight models is 91.10%. SAE seems particularly challenged for Los Angeles datasets with significantly lower accuracy achieved compared to other models (even dipping below 70% accuracy for g_i for I-Los). The main reason for this is that SAE model uses unsupervised learning method to reduce feature dimensions to obtain important hidden information instead of original data. Furthermore, we see a general trend of improved accuracy achieved when pre-processed datasets are used. For instance, we see a 12.57% improvement in accuracy for SAE for PI-Los compared to I-Los. This implies de-noised data offer better input for short-term traffic predictions.

Table 3 and Table 4 respectively present the prediction results of stations e_i and g_i. We observe cases where CLTFP achieves lower prediction error in terms of MAE and RMSE when compared to ALLSCP. Our results thus suggest

[2] We omit the plot for g_i as it is qualitatively similar.

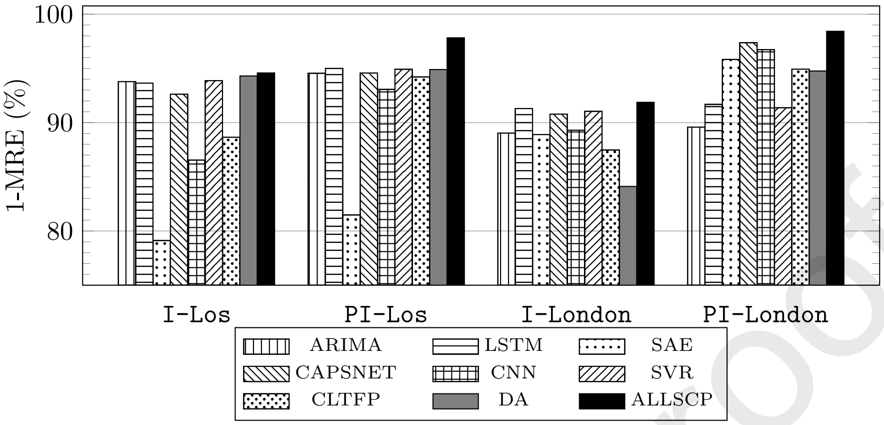


Figure 9: $(1 - MRE)\%$ for e_i on `I-Los` (left), on `PI-Los` (middle-left), on `I-London` (middle-right), and on `PI-London` (right). ALLSCP achieves the best accuracy for four cases.

that CLTFP is capable of reducing errors in absolute terms while appears to be less accurate with relative errors when ALLSCP performs better. This could be due to the fact that for CLTFP ensemble model, it exploits both LSTM and CNN models as constituent models while for ALLSCP, we proposed to use CAPSNET in place of CNN which as prior mentioned represents features in vector form rather than scalar values.

4.4.3. Ablation Experiments

To gain further insights into our ALLSCP model, we conducted ablation experiments in which we evaluate its performance by systematically removing individual module one at a time and conduct a comparison study. In total, five ALLSCP variants are tested, namely:

1. LLSCP – removal of the ARIMA module for short-term temporal feature analysis,

2. A-LSCP – removal of the LSTM module for medium-term temporal feature analysis,

3. AL-SCP – removal of the LSTM for long-term temporal feature analysis,

Table 3: Comparison of all models for the intersection $f_t^{e_i}$

Model	$f_t^{e_i}$ (`I-Los`)			$f_t^{e_i}$ (`PI-Los`)			$f_t^{e_i}$ (`I-London`)			$f_t^{e_i}$ (`PI-London`)		
	MAE	MRE	RMSE	MAE	MRE	RMSE	MAE	MRE	RMSE	MAE	MRE	RMSE
ARIMA	66.11	0.0622	92.42	59.36	0.0545	87.75	80.85	0.1097	126.93	73.04	0.1042	114.03
LSTM	64.57	0.0636	87.17	54.16	0.0501	79.78	66.17	0.0871	111.02	65.78	0.0832	104.48
SAE	169.22	0.2088	277.82	150.86	0.1852	221.46	83.78	0.1102	143.76	26.82	0.0418	40.68
CAPSNET	77.79	0.0738	103.03	58.12	0.0543	73.00	72.47	0.0922	117.88	20.68	0.0263	29.24
CNN	126.52	0.1346	225.44	61.76	0.0694	89.86	81.92	0.1070	134.18	25.44	0.0327	35.83
SVR	63.99	0.0613	90.74	54.52	0.0508	80.40	67.77	0.0896	112.34	66.67	0.0863	105.32
CLTFP	**32.84**	0.1135	**43.59**	**19.72**	0.0579	**27.71**	89.24	0.1253	145.48	34.01	0.0507	46.46
DA	61.00	0.0570	83.20	55.32	0.0511	81.63	75.56	0.1589	119.14	21.32	0.0524	30.14
ALLSCP	58.91	**0.0542**	79.90	24.28	**0.0218**	31.49	**63.23**	**0.0813**	**105.09**	**13.31**	**0.0158**	**18.68**

Table 4: Comparison of all models for the intersection $f_t^{g_i}$

Model	$f_t^{g_i}$ (`I-Los`)			$f_t^{g_i}$ (`PI-Los`)			$f_t^{g_i}$ (`I-London`)			$f_t^{g_i}$ (`PI-London`)		
	MAE	MRE	RMSE	MAE	MRE	RMSE	MAE	MRE	RMSE	MAE	MRE	RMSE
ARIMA	45.09	0.0843	61.33	36.56	0.0676	49.93	37.85	0.1035	52.13	34.10	0.0899	51.86
LSTM	42.04	0.0779	57.19	33.67	0.0631	45.78	30.99	0.0829	43.34	26.65	0.0798	41.69
SAE	157.42	0.3194	228.09	91.84	0.1937	164.81	36.01	0.0979	49.25	34.10	0.0946	46.44
CAPSNET	62.79	0.1186	82.73	27.56	0.0476	34.06	34.03	0.0873	47.05	11.94	0.0295	16.02
CNN	82.85	0.1631	125.16	32.75	0.0639	46.74	38.17	0.0977	52.96	12.44	0.0309	20.32
SVR	43.62	0.0838	60.84	33.77	0.0634	45.86	30.92	0.0831	43.00	29.10	0.0816	41.35
CLTFP	**21.63**	0.1204	**29.30**	11.07	0.0603	15.10	45.43	0.1186	63.11	21.71	0.0627	28.77
DA	40.69	0.0746	54.88	33.00	0.0608	44.66	35.19	0.0960	48.76	32.47	0.0931	47.01
ALLSCP	40.26	**0.0740**	54.73	**9.38**	**0.0171**	**12.45**	**29.09**	**0.0761**	**40.60**	**7.07**	**0.0171**	**9.80**

4. ALLCP – removal of the SAE for global spatial feature analysis and

5. ALLS – removal of the CAPSNET for local spatial feature analysis.

Table 5 shows results of the ablation experiments on the two datasets (`L-Los` and `L-London`) from linear roadways. Overall, the full-fledged ALLSCP model outperforms its five variants (i.e., removal of any sub-modules from ALLSCP negatively impacts its performance. On `L-Los`, LLSCP performs the worst. This indicates that the ARIMA used for short-term temporal feature analysis plays a critical role in our ALLSCP. From the table, we also note that CAPSNET also contribute heavily to the prediction accuracy as removing it causes the ALLS variant to record second worst predictions. Among five variants, AL-SCP achieves the best results, followed by A-LSCP and ALLCP. This indicates that, for short-term traffic prediction problem addressed in this paper, LSTM for long-term temporal feature analysis is less important than the other four modules. Similar results can be found on `L-London`. Therefore, on linear roadways, the importance of the different constituent modules in ALLSCP can be ranked (from most to least) as follows: {ARIMA, CAPSNET, SAE, LSTM for medium-term temporal feature analysis, LSTM for long-term temporal feature analysis}. From this analysis, we can also see that the short-term temporal and local spatial features are two most important features for short-term traffic prediction on linear roadways.

Table 6 presents ablation experimental results on two datasets (`I-Los` and `I-London`) from intersections. All models show better results on intersections than on linear roadways, and our proposed model, ALLSCP, still achieves the best performance. The reason for this is that more features in TS_t from intersections can be used for improving prediction accuracy. Among five variants, ALLS obtains the worst results, which indicates that CAPSNET module, that is responsible of analysing local spatial features, takes most important position in ALLSCP. AL-SCP is the best variant and its performance is very close to ALLSCP, which means LSTM used for long-term temporal feature analysis is less important than other four modules. Overall, the importance ranking

Table 5: The results of ablation experiments on linear roadways.

Model	L-Los			L-London		
	MAE	MRE	RMSE	MAE	MRE	RMSE
LLSCP	83.08	0.0741	108.79	42.97	0.0723	107.89
A-LSCP	73.37	0.0664	97.02	38.57	0.0545	95.99
AL-SCP	71.71	0.0645	95.16	36.35	0.0496	95.11
ALLCP	73.66	0.0665	97.49	41.13	0.0556	100.26
ALLS	78.22	0.0693	104.68	57.78	0.0802	119.21
ALLSCP	**71.64**	**0.0614**	**95.08**	**36.29**	**0.0495**	**95.08**

of modules from most to less in our ALLSCP on intersections is {CAPSNET, ARIMA, SAE, LSTM for medium-term temporal feature analysis, LSTM for long-term temporal feature analysis}. It shows that the local spatial and short-term temporal features are two most important features for short-term traffic prediction on intersections, and the local and global spatial features become more important than on linear roadways.

4.4.4. Prediction Stability and Robustness

We have shown that ALLSCP is consistent in making the best prediction accuracy for different scenarios. Hence, ALLSCP behaves stably over the different prediction scenarios. We also observe that generally, models perform better using pre-processed datasets when compared to using raw traffic data. For instance, the average accuracy of ALLSCP improves from 94.46% to 97.83% on linear roadways and from 92.76% to 98.29% on intersections.

We proceed to compare the improvement of $(1 - MRE)\%$ between raw and pre-processed data. We present the results in Table 7. All models with convolutional layers (i.e., CNN, CLTFP, CAPSNET, ALLSCP) achieve improved accuracy after removing noises using HP filter. This is due to the use of convolutional operator that is commonly used to extract edge features in image recognition applications. In our case for traffic flow, the edge feature corresponds to the difference of traffic flow between two continuous time intervals.

Table 6: The results of ablation experiments on intersections.

Model	$f_t^{e_i}$ (`I-Los`)			$f_t^{g_i}$ (`I-Los`)			$f_t^{e_i}$ (`I-London`)			$f_t^{g_i}$ (`I-London`)		
	MAE	MRE	RMSE	MAE	MRE	RMSE	MAE	MRE	RMSE	MAE	MRE	RMSE
LLSCP	62.30	0.0582	84.09	44.16	0.0834	59.50	63.67	0.0872	106.46	31.46	0.0882	43.04
A-LSCP	62.84	0.0584	84.83	43.55	0.0818	59.27	65.13	0.0901	111.79	31.21	0.0852	42.67
AL-SCP	62.86	0.0584	85.09	43.35	0.0813	59.11	63.65	0.0869	108.43	31.26	0.0858	43.11
ALLCP	63.85	0.0593	85.88	43.71	0.0822	59.44	66.13	0.0909	113.57	31.23	0.0850	42.82
ALLS	66.12	0.0615	92.09	45.11	0.0851	61.21	77.93	0.1087	125.22	36.63	0.1056	50.45
ALLSCP	**58.91**	**0.0542**	**79.90**	**40.26**	**0.0740**	**54.73**	**63.23**	**0.0813**	**105.09**	**29.09**	**0.0761**	**40.60**

Sudden changes (e.g., due to traffic accidents) causes unbalanced learning [54] and negatively affects the prediction. Pre-processing data smooths and reduces the differences between two intervals to enable us to obtain better results. DA also achieves improvement since DA is fed with more detailed temporal information (e.g., hourly, daily and weekly).

Table 7: Accuracy improvement when comparing raw data against de-noised data for different models.

Model	L-Los	L-London	I-Los	I-Los	I-London	I-London	Variance
			(e_i)	(g_i)	(e_i)	(g_i)	
ARIMA	2.22	-1.47	0.77	1.67	0.55	1.36	1.38
LSTM	0.82	-3.49	1.35	1.48	0.39	0.31	2.83
SAE	1.93	-0.59	2.36	12.57	6.84	0.33	30.41
CAPSNET	4.52	2.37	1.95	7.10	6.59	5.78	3.92
CNN	4.82	4.49	6.52	9.92	7.43	6.68	3.22
SVR	2.44	-0.40	1.05	2.04	0.33	0.15	1.04
CLTFP	4.67	0.20	5.56	6.01	7.46	5.59	5.14
DA	4.13	1.82	0.59	1.38	10.65	0.29	12.81
ALLSCP	4.30	2.45	3.24	5.69	6.55	5.90	2.20

From Table 7, we see that when comparing using raw and de-noised data, ALLSCP is among the models achieving the best improvements (average improvement = 4.69%). CNN achieves the highest improvements when using de-noised datasets with an average improvement of 6.64% though as we have shown before, its accuracy is much worse than ALLSCP. Furthermore, ARIMA, LSTM and SVR only achieve minimal improvements (i.e., below 1%). Along with SAE, these models even achieve worse performance using de-noised datasets for linear roadways in London. In terms of prediction stability (from the perspective of variance of the prediction improvements), SVR is the most stable with lowest variance among all models (i.e., only 1.04). The second lowest variance is achieved by ARIMA (i.e., 1.38) followed closely by our ALLSCP with variance of 2.20. Although the variances of SVR and ARIMA on all datasets are lower than ALLSCP, the prediction accuracy of ALLSCP is consistently higher than those two models. Therefore, considering both the prediction accuracy and the variance of improvements on all datasets, our ALLSCP is more stable, robust and accurate.

5. Conclusions

In this paper, we present a novel ensemble model for addressing the problem of short-term traffic flow prediction; a problem that has received renewed attention due to the development of smart city visions. Taking into account four important elements: 1) high quality data, 2) detailed temporal features of different scales, 3) local and global spatial features and 4) ensemble model construction approach, our models exploits the strengths of four submodels, namely ARIMA, LSTM, SAE and CAPSNET to make our predictions. We examine our proposed model, ALLSCP, across two different road types (linear and intersections) at two different locations (Los Angeles and London) where frequent congestion and accidents are expected. We also used both raw traffic data as well as pre-processed (i.e., de-noised) data. We compare our ALLSCP against existing models in the literature including its constituent submodels, two single models (namely SVR and CNN) and two existing ensemble models in the literature (namely DA and CLTFP). Our ALLSCP model achieved the highest accuracy among the nine considered models, achieving an average of 96.14% and 95.53% accuracy for linear and intersection roadways respectively while on average, the other competing models achieved 93.10% and 91.10% for the corresponding scenarios. Our results show that ALLSCP model is not only accurate but also the most robust, recording the least accuracy degradation when making predictions for the more challenging data with frequent congestion and accidents.

References

[1] W. G. Fink, Intelligent transportation systems, in: Microwave and Millimeter-Wave. Monolithic Circuits Symp., IEEE, 1995, p. 3.

[2] J. Zhang, et al., Data-driven intelligent transportation systems: A survey, IEEE Trans. Intell. Transp. Syst. 12 (4) (2011).

[3] B. Smith, Forecasting freeway traffic flow for intelligent transportation systems application., Transp. Res. Part A 1 (31) (1997) 61.

[4] E. I. Vlahogianni, M. G. Karlaftis, J. C. Golias, Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach, Transp. Res. Part C Emerg. Technol. 13 (3) (2005) 211–234.

[5] Y. Lv, et al., Traffic flow prediction with big data: A deep learning approach., IEEE Trans. Intell. Transp. Syst. 16 (2) (2015).

[6] G. E. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, Time series analysis, John Wiley & Sons, 1970.

[7] M. S. Ahmed, A. R. Cook, Analysis of freeway traffic time-series data by using Box-Jenkins techniques, no. 722, 1979.

[8] M. Van Der Voort, M. Dougherty, S. Watson, Combining kohonen maps with arima time series models to forecast traffic flow, Transp. Res. Part C Emerg. Technol. 4 (5) (1996) 307–318.

[9] S. Lee, D. B. Fambro, Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting, Transp. Res. Record 1678 (1) (1999) 179–188.

[10] B. M. Williams, L. A. Hoel, Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results, Journ. Transp. Eng. 129 (6) (2003) 664–672.

[11] G. P. Zhang, Time series forecasting using a hybrid arima and neural network model, Neurocomputing 50 (2003) 159–175.

[12] Y. Liu, Z. Zhang, J. Chen, Ensemble local kernel learning for online prediction of distributed product outputs in chemical processes, Chemical Engineering Science 137 (2015) 140–151.

[13] Y. Liu, C. Yang, Z. Gao, Y. Yao, Ensemble deep kernel learning with application to quality prediction in industrial polymerization processes, Chemometrics and Intelligent Laboratory Systems 174 (2018) 15–21.

[14] Z. Costello, H. G. Martin, A machine learning approach to predict metabolic pathway dynamics from time-series multiomics data, NPJ systems biology and applications 4 (1) (2018) 1–14.

[15] Z. Chen, Y. Liu, S. Liu, Mechanical state prediction based on lstm neural netwok, in: 2017 36th Chinese Control Conference (CCC), IEEE, 2017, pp. 3876–3881.

[16] K. Y. Chan, T. S. Dillon, J. Singh, E. Chang, Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenberg–marquardt algorithm, IEEE Trans. Intell. Transp. Syst. 13 (2) (2011) 644–654.

[17] K. Kumar, M. Parida, V. Katiyar, Short term traffic flow prediction for a non urban highway using artificial neural network, Procedia-Social and Behavioral Sciences 104 (2013) 755–764.

[18] K. Kumar, M. Parida, V. Katiyar, Short term traffic flow prediction in heterogeneous condition using artificial neural network, Transport 30 (4) (2015).

[19] B. Ghosh, B. Basu, M. O'Mahony, Bayesian time-series model for short-term traffic flow forecasting, Journ. Transp. Eng. 133 (3) (2007) 180–189.

[20] F. G. Habtemichael, M. Cetin, Short-term traffic flow rate forecasting based on identifying similar traffic patterns, Transp. Res. Part C Emerg. Technol. 66 (2016) 61–78.

[21] Y. Zhang, Y. Zhang, A. Haghani, A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model, Transp. Res. Part C Emerg. Technol. 43 (2014) 65–78.

[22] M.-C. Tan, et al., An aggregation approach to short-term traffic flow prediction, IEEE Trans. Intell. Transp. Syst. 10 (1) (2009).

[23] A. Stathopoulos, M. G. Karlaftis, A multivariate state space approach for urban traffic flow modeling and prediction, Transp. Res. Part C Emerg. Technol. 11 (2) (2003) 121–135.

[24] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, L. D. Han, Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions, Expert Syst. with App. 36 (3) (2009) 6164–6173.

[25] Z. Zhao, et al., Lstm network: a deep learning approach for short-term traffic forecast, IET Intell. Transp. Syst. 11 (2) (2017) 68–75.

[26] N. G. Polson, V. O. Sokolov, Deep learning for short-term traffic flow prediction, Transp. Res. Part C 79 (2017) 1–17.

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

[28] L. Li, S. He, J. Zhang, B. Ran, Short-term highway traffic flow prediction based on a hybrid strategy considering temporal–spatial information, Journ. Adv. Transp. 50 (8) (2016) 2029–2040.

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

[30] Y. Wu, et al., A hybrid deep learning based traffic flow prediction method and its understanding, Transp. Res. Part C Emerg. Technol. 90 (2018) 166–180.

[31] H. Zheng, F. Lin, X. Feng, Y. Chen, A hybrid deep learning model with attention-based conv-lstm networks for short-term traffic flow prediction, IEEE Transactions on Intelligent Transportation Systems (2020).

[32] T. Ma, C. Antoniou, T. Toledo, Hybrid machine learning algorithm and statistical time series model for network-wide traffic forecast, Transportation Research Part C: Emerging Technologies 111 (2020) 352–372.

[33] R. Yao, W. Zhang, L. Zhang, Hybrid methods for short-term traffic flow prediction based on arima-garch model and wavelet neural network, Journal of Transportation Engineering, Part A: Systems 146 (8) (2020) 04020086.

[34] X. Chen, H. Chen, Y. Yang, H. Wu, W. Zhang, J. Zhao, Y. Xiong, Traffic flow prediction by an ensemble framework with data denoising and deep learning model, Physica A: Statistical Mechanics and its Applications 565 (2021) 125574.

[35] G. Zheng, W. K. Chai, V. Katos, An ensemble model for short-term traffic prediction in smart city transportation system, in: IEEE Global Commun. Conf. (GLOBECOM), 2019.

[36] S. Blandin, A. Salam, A. Bayen, Individual speed variance in traffic flow: analysis of bay area radar measurements, in: Transportation Research Board 91st Annual Meeting, 2012.

[37] N. Zhao, et al., A practical method for estimating traffic flow characteristic parameters of tolled expressway using toll data, Procedia-Social and Behavioral Sciences 138 (2014) 632–640.

[38] M. Van Aerde, H. Rakha, Multivariate calibration of single regime speed-flow-density relationships, in: Proc. 6th Int'l. Conf. on Vehicle Navigation & Information Systems (VNIS), IEEE, 1995, pp. 334–341.

[39] C. Company, An introduction to the california department of transportation performance measurement system (pems).
URL http://pems.dot.ca.gov/

[40] Highways england network journey time and traffic flow data.
URL https://data.gov.uk/dataset/9562c512-4a0b-45ee-b6ad-afc0f99b841f/highways-england-network-journey-time-and-traffic-flow-data

[41] L.-C. Ma, W.-L. Xu, D.-X. Liu, Prediction model of traffic flow along typical roads in city urban district based on wavelet transform, Control and Decision 26 (5) (2011) 789–793.

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

[43] L. Jensen, Guidelines for the application of arima models in time series, Research in nursing & health 13 (6) (1990) 429–435.

[44] X. Ma, et al., Long short-term memory neural network for traffic speed prediction using remote microwave sensor data, Transp. Res. Part C Emerg. Technol. 54 (2015) 187–197.

[45] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780. doi:http://www.bioinf.jku.at/publications/older/2604.pdf.

[46] ufldl.stanford.edu, Stacked autoencoder.
URL http://ufldl.stanford.edu/wiki/index.php/Stacked_Autoencoders

[47] S. Menard, Applied logistic regression analysis, Vol. 106, Sage, 2002.

[48] R. Dunne, N. Campbell, On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function, in: Proc. Aust. Conf. on the Neural Networks, 1997.

[49] R. Everaers, K. Kremer, A fast grid search algorithm for molecular dynamics simulations with short-range interactions, Computer Physics Communications 81 (1-2) (1994) 19–55.

[50] S. Sabour, N. Frosst, G. E. Hinton, Dynamic routing between capsules, in: Advances in neural information processing systems, 2017.

[51] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: the 3rd International Conference for Learning Representations, San Diego, 2015.

[52] A. Maravall, A. Del Rio, et al., Time aggregation and the Hodrick-Prescott filter, no. 0108, Banco de España, 2001.

[53] W. Huang, G. Song, H. Hong, K. Xie, Deep architecture for traffic flow prediction: deep belief networks with multitask learning, IEEE Trans. Intell. Transp. Syst. 15 (5) (2014) 2191–2201.

[54] J. Pang, et al., Libra r-cnn: Towards balanced learning for object detection, in: IEEE Conf. Comp. Vision & Pattern Recog., 2019.