Short-Term Traffic Flow Forecasting: A wide and deep approach with periodic feature selection

Martin Esugo, Qian Lu, Olivier Haas

***Abstract*— Accurate traffic flow forecasting can improve the effectiveness of traffic management schemes, leading to reduced traffic congestions, emissions, costs, and improved health and productivity of commuters. However, the complex non-linear, periodic, and dynamic spatial-temporal nature of traffic flow data makes traffic forecasting challenging. This work proposes a deep-learning model, based on wide and deep architecture, to mine both periodic and spatial-temporal features. The wide section consists of a fully connected layer optimized to mine the periodic feature, while the deep portion consists of a conv-LSTM module, which extracts the spatial-temporal features. The network is made more efficient by analyzing traffic data periodicity and selecting the most relevant features, namely the weekly pattern. Simulation results demonstrate that our proposed model outperforms existing approaches whilst requiring one less input, thereby reducing the amount of pre-processing required. The evaluation of model accuracy considers the GEH statistic, used by traffic modelers, as well as traditional criteria adopted in traffic forecasting.**

***Index Terms*—Correlation analysis, Deep learning, Traffic flow forecasting, Conv-LSTM, Hyperparameter optimization.**

I. Introduction

Short-term traffic forecasting has been a major topic of interest since the 1980s [1] and has become an integral part of intelligent transportation systems (ITS). Accurately predicting future traffic states can alleviate traffic congestions by providing commuters with traffic information that helps in route selection, which in turn improves commuter journey times and travel time prediction whilst mitigating other negative effects such as environmental pollution [2] and revenue losses [3].

The goal of short-term traffic forecasting is to provide an accurate prediction of the future traffic states within a time horizon ranging from 5 min to 60 min [4], [5]. The accuracy of current prediction has been found in [6] to be on average 17% in terms of Mean Absolute Percentage Error (MAPE) between the projected forecasts and the true traffic volumes based on complied traffic forecast accuracy data from 1291 projects in both Europe and the United States. Surprisingly, whilst standard in traffic modelling, GEH and other traffic modelling criteria are not used to evaluate forecasting model performance [4], [5].

According to recent traffic forecasting surveys [5], [7], [8], traffic forecasting models fall into one of two distinct categories: statistical models or machine learning models. The machine learning category is further sub-divided into traditional machine learning models and deep learning models.

Statistical models include Kalman filtering techniques [9], [10] and time-series based methods including Historical Average (HA), Vector Auto-Regressive (VAR) [11], Auto-Regressive Integrated Moving Average (ARIMA) [12], [13], Seasonal ARIMA (SARIMA) [12], Bayesian networks [14], [15] networks. More details can be found in the review papers [16], [17]. Bayesian networks and ARIMA models were shown to have similar prediction performance [18], however, a linear conditional Gaussian Bayesian network proposed by the same authors [18] was shown to outperform various ARIMA models for 5, 10, and 15 min forecast horizons. A comparison between HA, VAR and a regression neural network for traffic flow prediction on the U.S. Highway 290 in Houston, Texas [19] demonstrated that the VAR model was superior in a case where there is little to no missing data and that the HA model had the worst prediction performance in terms of accuracy. According to [5], [17], most statistical models only consider the temporal dependence of traffic data, which limits their prediction accuracy. Advanced statistical techniques which incorporate spatial information e.g., space-time ARIMA (STARIMA) can overcome this limitation.

Traditional machine learning techniques applied to the traffic forecasting problem include K-nearest neighbors (KNNs) [20]–[22], support vector regression (SVR) [23], [24], and artificial neural networks (ANNs) [25]–[27]. Five different machine learning models which included two artificial neural networks, two support vector machines, and a multilinear regression model were compared with three statistical methods: ARIMA, VAR and a space-time (ST) model in [28]. The dataset used was collected from three links on the 4th ring road in Beijing. It was found that the machine learning methods outperformed statistical models for forecast horizons less than 5-timesteps ahead, with a 2 min sampling frequency. Unexpectedly, the ST model outperformed the machine learning models for longer forecast horizons. The authors suggest that this is due to the model incorporating both spatial and temporal information from adjacent stations. In [29], three different traffic prediction models were compared. The results were mixed for short-term forecasting, however, the ANN outperformed auto-regressive moving average (ARMA), and spatiotemporal-ARMA for long-term time horizons – above 60 min. used both statistical and machine learning methods were used in [18] to predict traffic flow for an urban road network in Shanghai. Machine learning models were found to be superior to statistical models. In general machine learning methods have gained more attention in recent years and traditional machine learning methods tend to outperform simple statistical

approaches but give mixed results when compared with more advanced statistical models [17].

The advent of deep learning-based models has further improved the effectiveness of ANNs in traffic forecasting [5], [30]. Deep learning models have a multi-layer architecture with many hidden layers and can thus potentially extract more features from complex data such as the spatial-temporal traffic data leading to an improvement in prediction performance [5]. Examples of deep learning models proposed for traffic forecasting include Deep Belief Network (DBN) [31], Stacked Autoencoders (SAE) [30], Long Short-Term Memory networks (LSTMs) [32], [33], and Convolutional Neural Networks (CNNs) [34], [35]. LSTMs are good at temporal feature extraction whilst CNNs are good for spatial feature extraction. This has led to their combination in hybrid deep learning (CNN-LSTM) models which combines the benefits of both the convolution and LSTM blocks [36]–[38], with the output fusion done either via an attention mechanism [38] or by concatenation [39]. A deep neural network (DNN) based model (DNN-BTF) [36] which consists of a series of one-dimensional (1D) convolutional and LSTM modules to mine the spatial, temporal and the weekly and daily periodicity showed good performance when compared to other traditional and some deep learning models. An attention-based Conv-LSTM cascade model coupled with a Bi-LSTM module for periodic feature mining was proposed in [37] for short-term traffic forecasting. Their results show that this model outperformed existing approaches including the DNN-BTF model especially when the periodic features were incorporated into the model. Overall, the literature suggests that hybrid deep learning models offer the best performance due to their ability to simultaneously mine both spatial, temporal, and periodic traffic features. Meanwhile, according to [5], [19], there is a need to identify the relevant periodic features – daily, weekly, and sometimes monthly and yearly – as this can lead to further improvement in model performance. However, there is a lack of research exploring this area of interest.

Motivated by this, a rigorous method is presented in this paper to select the most appropriate periodic traffic feature. The selected periodic feature and the spatial-temporal data are subsequently fed as inputs to a wide and deep Conv-LSTM neural network. The novelties of this paper are as follows:

- It extends the application of the hybrid wide and deep neural network originally proposed in [40], by employing a fully connected layer placed in the wide channel to memorize features extracted from the periodic data while a Conv-LSTM module is placed in the deep channel for spatial-temporal feature extraction. The advantage of this configuration is that the memorized periodic features from the wide channel can be used to refine the output of the deep network.
- It carries out a systematic statistical analysis based on Pearson's correlation coefficient to demonstrate that, for the two locations and the forecast horizons considered, daily patterns are not necessary whereas weekly patterns are the most appropriate long-term temporal feature for forecasting purposes.
- It uses two real-world traffic datasets to demonstrate the generalizability of the proposed approach and its superior performance compared to the published literature.
- To ensure a fair comparison, it optimizes all the hyperparameters for each network and each time horizon from 5 min to 60 min in the same manner.

The remainder of the paper is organized as follows. Section II presents the methodology, which starts with details of the dataset used, the data analysis and associated preprocessing steps taken. It then formulates the traffic forecasting problem. Section III presents the model. Section IV describes the training configuration and critically evaluates the performance of the model. Finally, Section V concludes the paper and offers recommendations for further work.

II. Methodology

A. Dataset description

Two urban freeway traffic flow datasets were collected from the California transportation agency performance management system (Caltrans-PeMS) [41]. The datasets, sampled every 5 min, include weekdays, weekends, and possible holidays.

The first dataset contains 35,136 samples representing 4 months (1st March 2019 - 30th June 2019) for a 10 km section of the West Side freeway in Stockton, California (district 10). The second dataset contains 53,280 samples over 5 months (1st September 2017 - 4th March 2018) for a 2.3 km section of the John B. Williams freeway in Oakland, California (district 4). The detector selection process for each stretch of urban freeway is carried out as follows. First, detectors with more than 2000 missing samples are dropped. Second, a spatial correlation analysis is used to select the most appropriate detectors where any detector with a spatial correlation less than 0.85 relative to the other locations is removed. This selection process aims to reduce the number of detectors on both freeways to focus solely on the most reliable data. In this work, the number of detectors is reduced from 10 to 7 for the Stockton dataset, and from 11 to 6 for the Oakland dataset. Fig. 1 shows the distribution of selected detectors across the freeway in both the Stockton (left) and Oakland (right) datasets. The unlabeled black pins indicate the position of detectors not selected for use. Each labelled pin represents the location of a detector from where the traffic data is collected. The position S_0 represents the randomly selected point of interest (POI), that is, the point where traffic is to be predicted. Fig. 2 is a heatmap that shows the traffic flow per day for the final five weeks of each dataset. It is observed that there exist two peaks in the Stockton dataset, a narrow morning and a more spread-out evening rush hour. There is only one evening peak in the Oakland dataset. A weekly periodicity of the traffic flow is observed in both datasets, however, more variability is observed in the Oakland dataset over the 5 months duration. The last two weeks of each dataset (4032 samples) are used for testing, while the remainder is used for training the proposed model. The red bar in Fig. 2 indicates the start of the test set in each dataset.

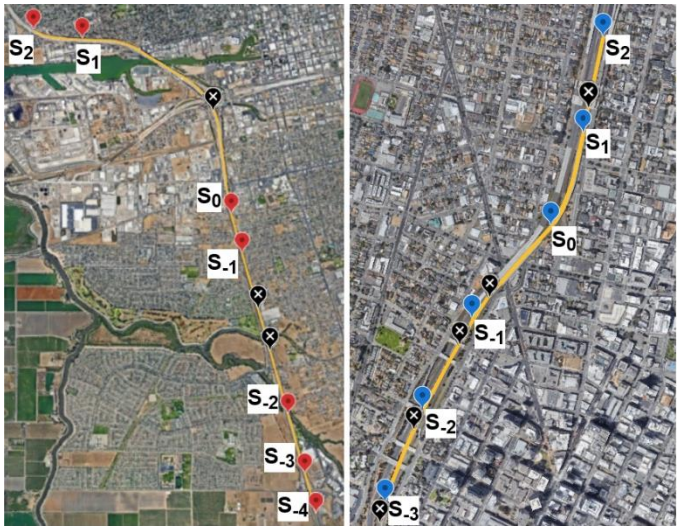

Fig. 1. The road section of the Stockton (left) and Oakland (right) datasets where S_0 represents the location where the traffic is predicted.

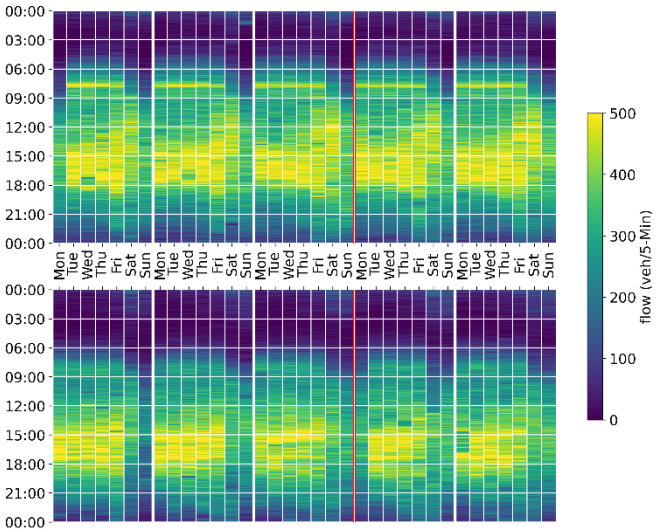

Fig. 2. Heatmap of the traffic flow for the final five weeks: Stockton (top), Oakland (bottom)

B. *Data Analysis*

Traffic data shows both spatial-temporal correlations and periodicity characterized by long-term temporal correlations. The latter could include weekly (same day in previous weeks) or daily (different days in the same week) patterns. In this work, Pearson's correlation coefficient, denoted ρ, is adopted to select the most important long-term temporal feature. ρ can be any value between -1 and $+1$. A value of 0 indicates that there is no correlation and values closer to -1 or $+1$ respectively indicate a strong negative or strong positive correlation. ρ is computed as follows:

$$\rho = \frac{\mathrm{cov}\left(f_x, f_y\right)}{\sigma_{f_x}\sigma_{f_y}} \tag{1}$$

where $cov(f_x, f_y)$ is the covariance between the traffic flow of one cycle f_x and a second cycle f_y, while σ_{f_x} and σ_{f_y} represent the standard deviation of the flow cycle f_x and f_y, respectively. To build the correlation matrices for both the daily and weekly patterns, the original traffic data of the POI is reshaped into a daily, working days (Mon. to Fri.), weekends (Sat. and Sun.) and weekly cycles before computing the correlation coefficients. Table I shows the minimum, mean, maximum and standard deviation of the correlation coefficient. Note that these results remain consistent throughout the dataset, irrespective of the weeks selected. The weekly data exhibit the highest correlation, closely followed by 'working days' with weekend and daily data being less correlated and characterized by the higher spread of the degree of correlation as demonstrated by higher standard deviation. The higher standard deviation in the degree of correlation for the daily pattern reflects the significant difference between traffic during the working days and weekends, see Fig. 2. Considering the working day data and weekend data as separate inputs would increase the model complexity. However, the correlation analysis has demonstrated that for the selected dataset, considering solely the weekly pattern is better as it is more consistent and therefore offers a more reliable long-term temporal feature.

TABLE I
LONG-TERM TEMPORAL CORRELATION DATA

Data	Index	Daily	Working Week	Weekend	Weekly
Stockton	**Min.**	**0.73**	**0.90**	**0.73**	**0.92**
	Mean	0.91	0.96	0.96	0.97
	Max.	0.98	0.98	0.94	0.98
	SD	**0.06**	**0.03**	**0.04**	**0.02**
Oakland	**Min.**	**0.78**	**0.88**	**0.78**	**0.90**
	Mean	0.92	0.95	0.96	0.96
	Max.	0.98	0.98	0.94	0.98
	SD	**0.05**	**0.03**	**0.04**	**0.03**

C. *Problem formulation*

This paper approaches the traffic forecasting problem from a supervised learning perspective. It maps the input variables – previous and current traffic flow variables – to the output variable – future traffic flow variable – using the sliding window method [4], [42].

First, the short-term spatial-temporal data is prepared as an input to the proposed model. By applying the aforementioned method, this input becomes the traffic flow f for each selected location $S = \{s_a, \dots, s_0, \dots, s_b\}$ during the time interval $t = \{k, k-1, k-2, \dots, k-n_r\}$; where s_a and s_b represents the first and last detector location upstream and downstream of s_0 respectively; k is the current time step and n_r is the number of lag observations considered. Combining the historical traffic flow of each selected location and time interval generates the short-term spatial-temporal input, denoted F_t^s, expressed as follows:

$$F_t^s := \begin{bmatrix} f_{k-n_r}^{s_a} & \cdots & f_{k-n_r}^{s_0} & \cdots & f_{k-n_r}^{s_b} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ f_{k-1}^{s_a} & \cdots & f_{k-1}^{s_0} & \cdots & f_{k-1}^{s_b} \\ f_{k}^{s_a} & \cdots & f_{k}^{s_0} & \cdots & f_{k}^{s_b} \end{bmatrix} \tag{2}$$

Second, the long-term temporal data is prepared as a second

input to the proposed model using the same sliding window technique and the same output as a reference point. As discussed in Section B, the long-term temporal input is best modelled by the weekly periodicity. From table I, it is observed that the traffic flow at a detector location for any given point in time is highly correlated with the traffic flow at the same location and same time step in the previous weeks. To account for inter-week traffic flow variability, a time step tolerance is added to include adjacent time steps from the past week. The resulting matrix, $F_w^{s_0}$, represents the long-term temporal input at location s_0 for each preceding week w considered, it is given by:

$$F_w^{s_0} := \begin{bmatrix} f^{s_0}_{k_w^c - n_k} & f^{s_0}_{k_w^c - (n_k - 1)} & \cdots & f^{s_0}_{k_w^c} & \cdots & f^{s_0}_{k_w^c + n_k} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ f^{s_0}_{k_w^2 - n_k} & f^{s_0}_{k_w^2 - (n_k - 1)} & \cdots & f^{s_0}_{k_w^2} & \cdots & f^{s_0}_{k_w^2 + n_k} \\ f^{s_0}_{k_w^1 - n_k} & f^{s_0}_{k_w^1 - (n_k - 1)} & \cdots & f^{s_0}_{k_w^1} & \cdots & f^{s_0}_{k_w^1 + n_k} \end{bmatrix} \quad (3)$$

where k_w^1 represents the same moment in time in the previous week (2016 samples behind the output), k_w^2 represents the same moment in time two weeks ago (2×2016 samples behind the output) and k_w^c represents the same moment in time c weeks ago (c×2016 samples behind the output). n_k is the number of adjacent time samples considered on either side of the time k_w^c. n_k is introduced to accommodate the time shift and variation in the current traffic flow compared to the flow observed in previous weeks.

Given that the location s_0 is considered as the POI, the output becomes the traffic flow value at s_0 at the time horizon $(k + n_p)$; where n_p is an integer value that represents the prediction horizon. This output represents the forecast, and it is denoted $F_{k+n_p}^{s_0}$. Similar to [38], this work adopts a 75 min ($n_r = 15$), lag observation to predict 5, 15, 30 and 60 min ahead, corresponding to $n_p = 1, 3, 6, 12$. In this work, the most appropriate parameters for the weekly periodic input n_k and c were found to be 3 and 2, respectively. This can be interpreted as using the same time interval from the previous two weeks, ± 15 min from each preceding week as a long-term temporal input.

D. Data preprocessing

Raw traffic data usually contains some erroneous and missing values which can be caused by several reasons, including transmission distortion, temporary power failure, unscheduled maintenance of the system and sensor fault. This paper adopts the following missing data imputation techniques depending on the type of missing data.

1) *Missing Completely at Random (MCR)*: missing data points are non-sequential, independent of one another and are situated between two known values. These missing data points are replaced with the average of the preceding and succeeding time steps.

2) *Missing at Random (MR)*: in this case, missing data points follow each other sequentially, and these long sequences are randomly distributed throughout the dataset. The nature of this type of missing value renders the simple interpolation method insufficient, thus a matrix-based missing data representation [44] in conjunction with the temporal day pattern is used to re-organize the data. The day pattern represents data collected at the same time step of the same day d, but in adjacent weeks. A matrix representation of a day pattern F_d^s, for a given spatial location S is given below:

$$F_d^s := \begin{bmatrix} f^s_{k_w^1} & f^s_{k^*} & f^s_{k_w^{-1}} \\ f^s_{k_w^1+1} & f^s_{k^*+1} & f^s_{k_w^{-1}+1} \\ \vdots & \vdots & \vdots \\ f^s_{k_w^1+n_m} & f^s_{k^*+n_m} & f^s_{k_w^{-1}+n_m} \end{bmatrix} \quad (4)$$

Here, the middle column represents a location S with a sequence $\{k^*, k^*+1, \ldots, k^*+n_m\}$ of missing traffic flow data; where k^* represents the start of the missing sequence, and n_m represents the length of the missing sequence. The first and last columns represent the same sequence of the same day, one week behind and one week ahead, respectively. After the generation of this matrix, the linear interpolation (i.e. the average) method is applied across the rows to fill in the missing data such that:

$$f^s_{k^*} = \frac{f^s_{k_w^1} + f^s_{k_w^{-1}}}{2} \quad (5)$$

In addition to missing data imputation, we also check for the presence of outliers using the interquartile range IQR, which measures the statistical dispersion of the data per spatial location with a more robust response to the presence of outliers. The IQR is given as the difference between the third $Q3$ and the first $Q1$ quartiles:

$$IQR = Q3 - Q1 \quad (6)$$

We define outliers as values which fall outside of the boundary conditions given by the upper boundary UB (7) and lower boundary LB (8).

$$UB = Q3 + 1.5 \times IQR \quad (7)$$

$$LB = Q1 - 1.5 \times IQR \quad (8)$$

On application of these boundary conditions, we conclude that our datasets do not contain outliers, therefore we proceed to the data scaling step without any modification to the data.

Data scaling is an important data pre-processing step for deep learning, as neural networks can be sensitive to the different scales in the data. Due to the varying number of lanes at each spatial location, the minimum, maximum and general flow values for each location differ. Thus, the Min-Max scaling method is used to normalize the data between 0 and 1. The normalization formula is given as follows:

$$f_k^{s*} = \frac{f_k^s - f^s_{\min}}{f^s_{\max} - f^s_{\min}} \quad (9)$$

where f_{min}^s and f_{max}^s represent the minimum and maximum flow values for a given spatial location S, respectively, f_k^s represents the current flow value of that spatial location that is

to be scaled, and f_k^{s*} is the output of the equation which represents the generated scaled value of f_k^s.

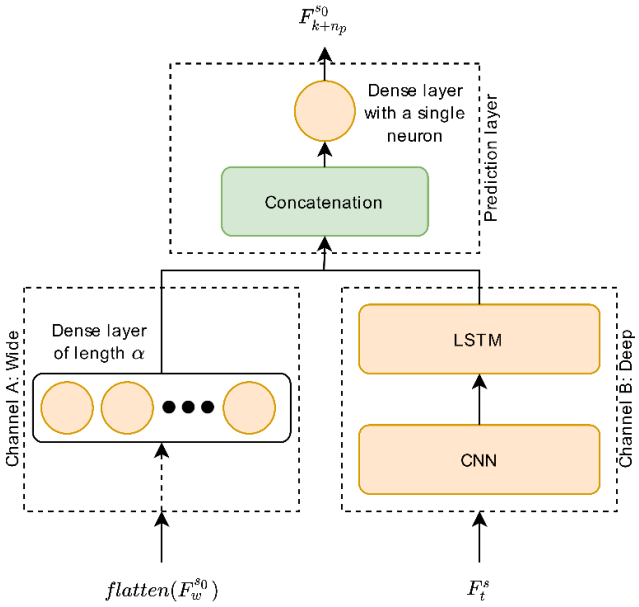

Fig. 3. Proposed model based on the wide and deep framework

III. Proposed Model

The proposed model makes use of the wide and deep framework originally proposed by [40]. As is illustrated in Fig. 3, a fully connected layer is placed in the wide channel and a Conv-LSTM module in the deep channel.

A. Wide Channel

The wide channel A consists of a fully connected layer of length α which is designed to learn global knowledge from the flattened weekly periodic matrix $F_w^{s_0}$. By using a fully connected neural network in the wide channel, we can memorize the periodicity in the traffic data and use the derived features of periodicity to augment the spatial features extracted in the deep channel [40], [43]. The output of each neuron in the fully connected layer is computed using the following equation:

$$y_j = \sum_{i=1}^{n} w_{i,j} x_i + b \tag{10}$$

Here, $w_{i,j}$ represents each neuron's weight between the input value i and the j^{th} neuron, with b being the bias term. n represents the length of the flattened 1D input data x. The output y_j is then passed through an activation function, which in conjunction with the number of neurons in the layer, determines its contribution to the feature map generated by the network. In this paper, the Rectified Linear Unit (ReLU) is used as the activation function due to its ability to prevent overfitting [44]. The ReLU activation gives a non-zero output only if the input is positive. It is defined as follows:

$$f(x) = \begin{cases} 0 & for \quad x < 0 \\ x & for \quad x \geq 0 \end{cases} \tag{11}$$

B. Deep Channel

The deep channel consists of a Conv-LSTM block, which is the combination of a Convolutional layer with an LSTM backend. This allows for the extraction of the features inherent in the short-term spatial-temporal traffic flow input F_t^s. The convolutional layer extracts the spatial features via a 2D convolutional operation which is performed over the data input F_t^s at each time step. By sliding a convolutional kernel or filter across the input, a feature map is obtained which can then be flattened and passed to the next layer. This process can be expressed as follows:

$$J_i = \sigma(w * F_t^s + b) \tag{12}$$

where w represents the weights of the convolutional kernel, F_t^s is the short-term spatial-temporal input, $*$ and σ represents the convolution operator and activation function, respectively. After the convolution operations are performed, a set of feature maps H is obtained:

$$H = \{J_1, J_2, ..., J_i\} \tag{13}$$

A pooling layer is not used here because the dimensionality of the input vector is already low. This output H of feature maps is passed as the input to the LSTM block. The LSTM block consists of several gates and memory units, which allow it to retain important information and discard unimportant ones. The process through which it does this is described as follows. First, the LSTM determines which information to retain or discard by passing the previous hidden state vector h_{t-1} and the cell input vector x_t through a forget gate f_t which contains a sigmoid activation σ that outputs a value between 0 and 1. Values close to 0 are discarded while values close to 1 are kept.

$$f_t = \sigma({w_{hf}}^T h_{t-1} + {w_{xf}}^T x_t + b_f) \tag{14}$$

The next step is to determine what new information is to be stored in the cell state. This is a two-part process that starts with the previous hidden state h_{t-1} and cell input x_t passed through a sigmoid and tanh layer respectively. The sigmoid layer determines the value to be updated and the tanh layer helps to regulate the network. Subsequently, the outputs from both activations are multiplied together and passed on to the next step.

$$i_t = \sigma({w_{hi}}^T h_{t-1} + {w_{xi}}^T x_t + b_i) \tag{15}$$

$$\tilde{C}_t = \tanh({w_{hC}}^T h_{t-1} + {w_{xC}}^T x_t + b_C) \tag{16}$$

The previous cell state C_{t-1} is updated to the next cell state C_t by firstly forgetting the unimportant information from the previous cell state via an element-wise multiplicative operation between C_{t-1} and f_t, then, generating the new cell state C_t via pointwise addition to the result of the element-wise product of i_t and $\tilde{C}_t$ from the previous step.

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \tag{17}$$

Finally, the output gate of the LSTM determines what the next hidden state should be. First, the previous hidden state and cell input (h_{t-1}, x_t) is passed through a sigmoid activation; then, the

newly generated cell state C_t from the previous step is passed through a tanh activation. Both results are combined using an element-wise product to determine the value for the new hidden state h_t. Both the new hidden state and the new cell state from the previous step are then carried over to the next time step.

$$o_t = \sigma(w_{ho}{}^T h_{t-1} + w_{xo}{}^T x_t + b_o) \quad (18)$$

$$h_t = o_t \otimes \tanh(C_t) \quad (19)$$

where $w_{hf}, w_{hi}, w_{hc}, w_{ho}$ and $w_{xf}, w_{xi}, w_{xc}, w_{xo}$ denote the respective weight matrices of each layer for the connection to the respective previous hidden state h_{t-1} and cell input x_t, and b_f, b_i, b_c, b_o represent the respective bias terms.

C. Prediction layer

The output of the wide channel A and deep channel B are concatenated, and the resulting vector is fed as an input to a fully connected neural network. This neural network consists of a single neuron that is densely connected to the preceding layer; that is, this layer is connected to every input from the concatenation layer. This layer uses no activation as the problem has been modelled as a regression problem and the value to be predicted is the numerical traffic flow with no transformation required. The output from this layer is the scaled prediction of the model which is subsequently unscaled to give the final prediction.

IV. Evaluation

A. Training setup

Keras, which is a deep learning API with a TensorFlow backend was used in the development of the model. Before the final training of the proposed model and all other models used for comparison, hyperparameters were optimized using the random search method. The random search method works by randomly combining a number of hyperparameters from a given hyperparameter space, with one random combination being a possible model configuration. The selection of hyperparameters tuned in the model is given in Table II. In setting up the random search, the optimization objective is set to minimize the validation loss. The number of trials and the number of executions per trial is set to 60 and 5 respectively. The former is selected based on the recommendation by [45] and the latter serves as a method to reduce result variance which occurs due to the random initialization of the network variables.

On completion of the optimization scheme, the best hyperparameters are selected and used in the final model training. The loss function selected is the mean squared error (MSE), defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (20)$$

where y_i represents the actual flow value, $\hat{y}_i$ represents the predicted flow values and n represents the total number of test samples.

The validation set is allocated 10% of the training data and an early call back with a patience of 5 epochs is used to monitor the validation loss and stop the training scheme when there is no more improvement in performance. To improve the robustness of the optimization against random parameter initialization, we run the training scheme 100 times and select the parameters based on the average value of each performance metric.

TABLE II
Hyperparameter Range

Hyperparameter	Range of values
Wide Channel	
Number of units (FC)	$[10..100, 1]$
Deep Channel	
Number of filters (CNN)	$[32..512, 32]$
Number of units (LSTM)	$[10..100, 1]$
Other	
Learning rate	$[1e^{-2}, 1e^{-3}, 1e^{-4}]$

B. Performance measures

To compare the efficacy of the proposed model to the current state of the art, the following standard performance metrics for traffic forecasting are employed: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and MAPE. These metrics are computed using the following equations:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (21)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (22)$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \quad (23)$$

where y_i represents the actual flow value, $\hat{y}_i$ represents the forecasted flow values and n represents the total number of test samples. A lower value for (21), (22) and (23) indicates a better performing model.

In addition, the GEH statistic [46] is computed for pairs of actual and forecasted flow values for each time interval and forecast horizon in the test set. This metric is non-linear, thus, one acceptance threshold can be applied to a wide range of traffic volumes using GEH. The computation for GEH is given as follows:

$$GEH = \sqrt{\frac{2\left(Y_i - \hat{Y}_i\right)^2}{Y_i + \hat{Y}_i}} \quad (24)$$

where $Y_i = y_i \times 12$ represents the actual hourly flow, and $\hat{Y}_i = \hat{y}_i \times 12$ represents the forecasted hourly flow. A GEH value less than 5 ($GEH < 5$) is considered a good fit, therefore the statistic of interest is the percentage of GEH values less than 5 computed for the entire test set. The acceptable value for modelling purposes is the percentage of $GEH > 85\%$ [46]. We believe that the percentage of GEH for forecasting should meet the same standards as that which is used for modelling. The

GEH applied to each data point sampled at 5 min is denoted GEH5. It is observed that traffic count exhibits small oscillations throughout the day. Assuming that these oscillations are not significant if they do not prevent significant peaks in traffic from being modelled, a moving average filter of dimension 3, representing a 15 min duration, is applied to smooth the measured and predicted flows. Padding is used to remove the time shift between original and smoothed data. GEH15 is calculated for the aforementioned smoothed data.

C. Model performance

A number of experiments were conducted to compare the performance of the proposed wide and deep model with two stand-alone models and a hybrid deep learning model. The POI for each dataset is set at the location S_0, and the final 2 weeks of each dataset are used as the test set for all the experiments. The two selected baseline models are the LSTM [32] and the CNN-LSTM [39] models. For the LSTM network, the number of layers is set to two, while the number of units per layer is optimized using the optimization scheme outlined in the training setup section. The CNN-LSTM network contains a single convolutional layer with an LSTM backend. The number of filters in the convolutional layer and the number of units in the LSTM layer are both optimized. For each experiment, the size of the input lag window is set to 15, meaning 75 min of historical data is used as input to the model. The attention-Conv-LSTM with double Bidirectional LSTM [37] is the hybrid deep-learning model selected for comparison. The main channel consists of two convolutional layers with two LSTM layers backend and an attention mechanism used for data fusion. In addition, it also consists of two side bi-LSTM layers which are used to extract features from the daily and weekly periodic patterns. The optimization scheme detailed in the training setup section is used to optimize the hyperparameters of this network before model evaluation. Fig. 4 and Fig. 5 show the prediction efficacy of the proposed model for different forecast horizons over a period of one day for the Stockton and the Oakland test dataset respectively. In each figure, the top subplot shows a comparison between the actual flow and the forecasted flow, while the bottom subplot shows the error given in terms of GEH. The black points indicate a forecast with acceptable GEH ($GEH5 < 5 \,\&\, GEH15 < 5$). The red points indicate a forecast with sub-optimal according to GEH5. The green points indicate a GEH which is sub-optimal for both the $GEH5$ and $GEH15$. The 15 min moving average gives a larger percentage of acceptable GEH (percentage GEH > 85%). This is due to the smoothing effect of the filter which reduces flow variations from neighboring time samples. The latter reduces the ability of the prediction model to track sudden variations whilst improving its ability to track trends. Note that, for the data sets considered, the main peaks in traffic are of a duration that is longer than 15 min, therefore using a smoothed version of the model output is appropriate.

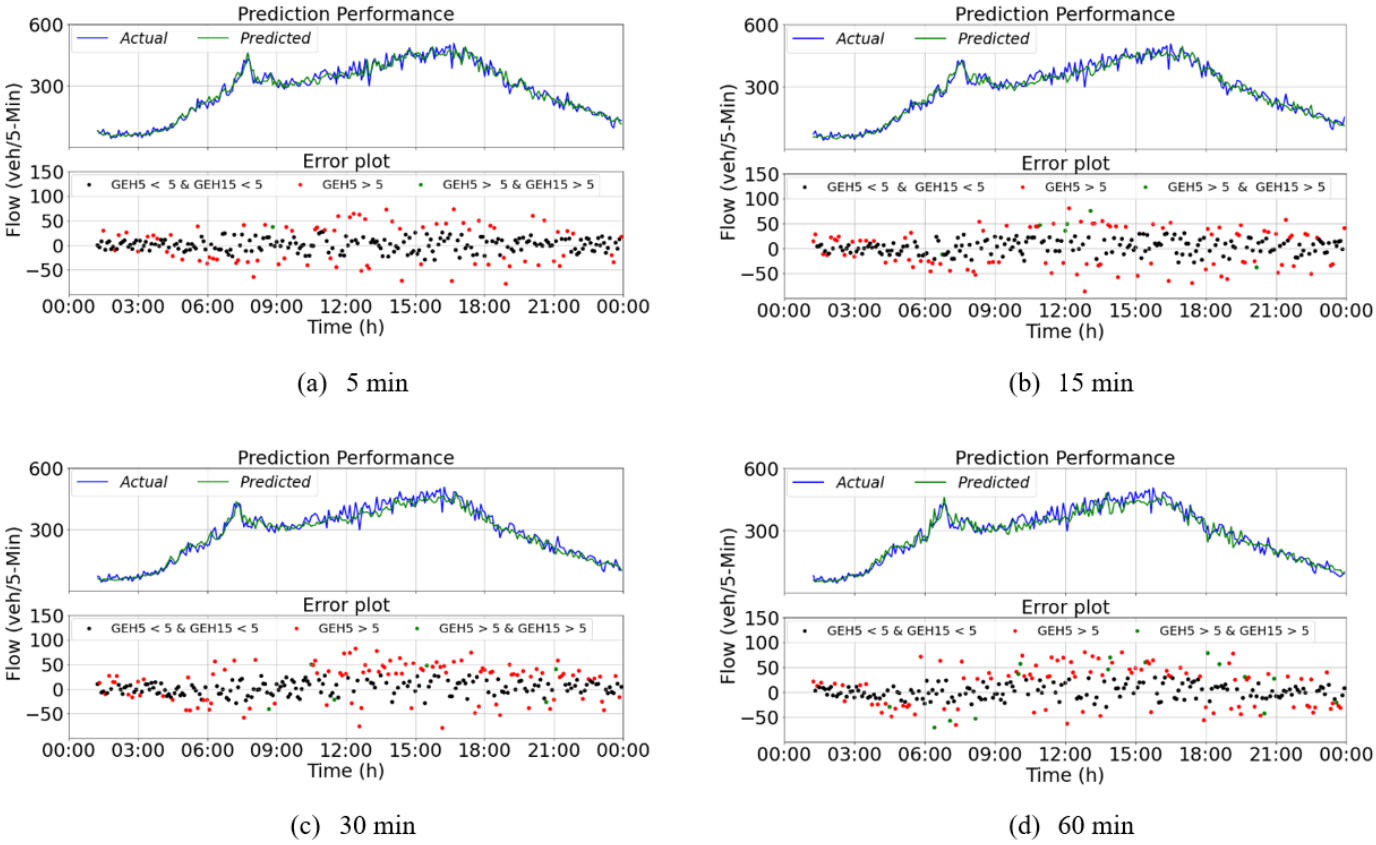

Fig. 4. *One-day prediction performance of the proposed model and associated error for different forecast horizons (Stockton)*

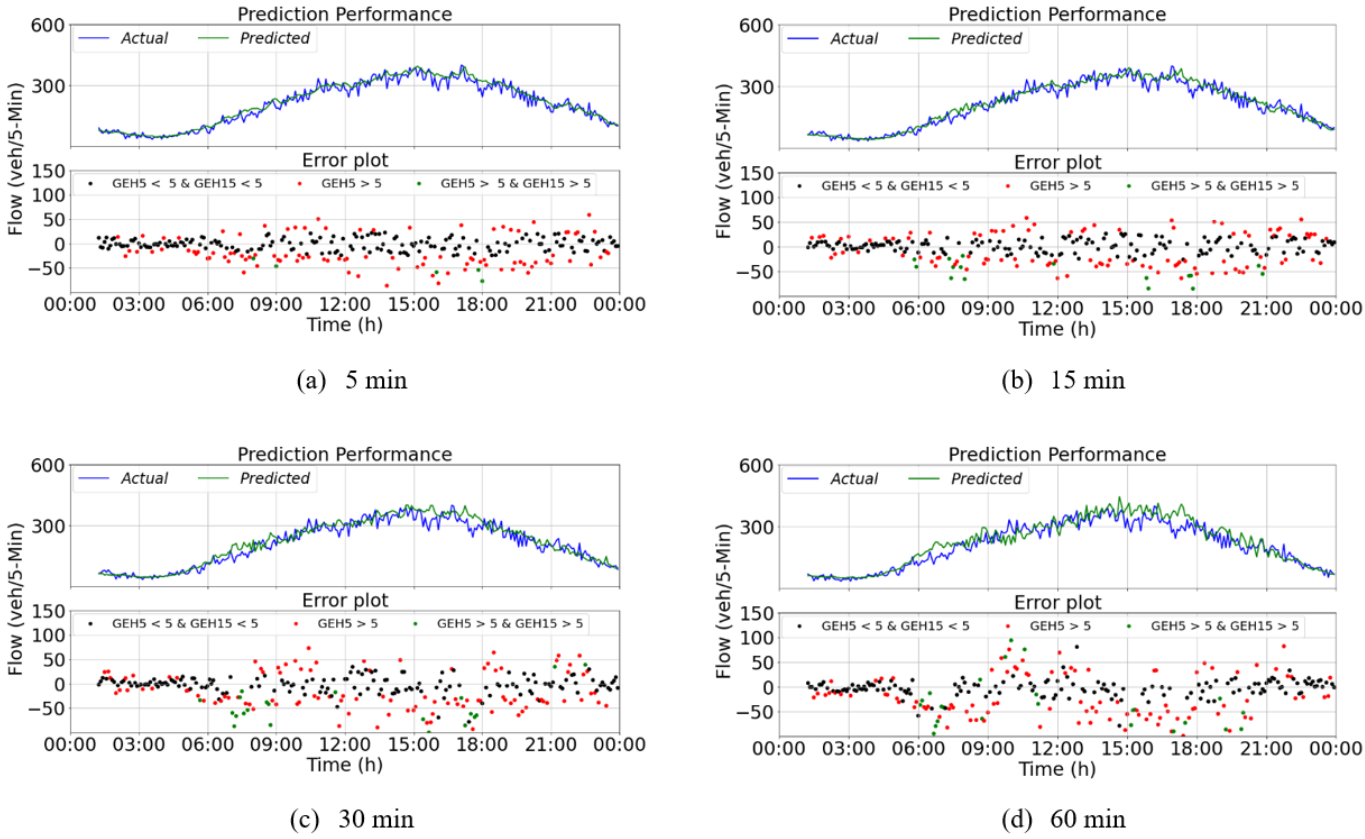

Fig. 5. *One-day prediction performance of the proposed model and associated error for different forecast horizons (Oakland)*

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT MODELS FOR STOCKTON NETWORK AND OAKLAND NETWORK

		Stockton				**Oakland**			
Model	Index	5 min	15 min	30 min	60 min	5 min	15 min	30 min	60 min
LSTM	MAE	17.8	25.63	27.47	28.52	22.09	26.17	26.52	30.38
	RMSE	23.49	33.2	36.68	37.37	28.98	35.73	36.43	44.37
	MAPE	7.76%	10.55%	11.95%	13.21%	10.98%	12.71%	13.21%	14.45%
	GEH_5	68.79%	54.34%	53.18%	51.05%	58.20%	55.76%	54.11%	48.65%
	GEH_{15}	99.4%	94.3 %	90.97%	88.55%	96.36%	93.3%	89.55%	84.72%
Conv-LSTM	MAE	20.87	24.49	26.59	29.51	22.57	25.42	25.86	29.6
	RMSE	27.8	32.47	35.16	39.16	31.03	35.34	35.87	41.93
	MAPE	8.88%	10.51%	11.35%	12.54%	11.06%	12.17%	12.28%	13.58%
	GEH_5	67.71%	58.88%	56.88%	52.72%	62.6%	56.67%	55.75%	52.07%
	GEH_{15}	99.37%	94.41%	91.16%	90.03%	98.17%	93.55%	90.42%	86.11%
AT-Conv-LSTM	MAE	18.07	22.98	23.91	24.19	21.99	25.0	25.76	28.52
	RMSE	23.85	30.05	31.35	32.44	29.69	34.45	35.04	39.64
	MAPE	7.81%	9.93%	10.29%	11.37%	10.87%	12%	12.17%	13.36%
	GEH_5	70%	58.11%	57.2%	57.33%	62.93%	57.48%	55.87%	53.03%
	GEH_{15}	98.21%	94. 67%	94.39%	90.88%	98.21%	93.07%	92.34%	89.21%
Wide & deep ConvLSTM	MAE	**18.12**	**21.47**	**22.94**	**23.33**	**19.82**	**24.76**	**25.3**	**28.3**
	RMSE	**23.93**	**28.04**	**30.02**	**30.61**	**26.78**	**34.56**	**34.47**	**39.5**
	MAPE	**7.86%**	**9.37%**	**9.85%**	**10.07%**	**9.85%**	**11.89%**	**12.16%**	**13.09%**
	GEH_5	**70.15%**	**63.94%**	**62.06%**	**56.94%**	**63.08%**	**57.58%**	**57.98%**	**55.74%**
	GEH_{15}	**99.73%**	**97.86%**	**97.28%**	**94.28%**	**98.31%**	**94.52%**	**92.32%**	**90.34%**

Table III shows the averaged prediction accuracy in terms of the selected performance metrics and GEH for both traffic networks over the selected test period. Compared to the vanilla LSTM model, the proposed wide and deep model shows similar performance for the 5 min Stockton forecast. However, at higher forecast horizons and for the Oakland dataset, the proposed model outperforms the vanilla LSTM. A similar observation is made when the proposed model is compared with the stand-alone Conv-LSTM model. Although the Conv-LSTM model can outperform the vanilla LSTM model in most cases,

it still falls short when compared to the proposed model. This improvement in the performance of the proposed model can be attributed to the model being able to capture more relevant periodic features in the wide channel which it combines with the spatial-temporal features mined by the Conv-LSTM block in the deep channel to provide for more accurate predictions. The AT-Conv-LSTM block was designed to capture both the spatial-temporal features and, the daily and weekly periodic features via its accompanying Bi-LSTM modules. However, from the results in tables III, the proposed wide and deep model outperforms the AT-Conv-LSTM model. The performance improvement can be attributed to the selection of the most relevant periodic feature undertaken as an extra data pre-processing step before constructing the model. It was determined via correlation analysis that the daily periodic data was not as relevant as the weekly periodic data, due to the relatively low correlation between weekdays and weekends.

Note that the relative error reduction achieved by the proposed model is small, being in the order of 1% for the Oakland network and ranging from -0.2% to 3% for the Stockton network. In the latter case, the vanilla LSTM model was marginally better for the 5 min prediction horizon but significantly worse than the other network for longer range predictions. This indicated that for a very short prediction horizon, it is sufficient to use immediately preceding flow measurements, whilst additional daily and weekly patterns are required to predict longer horizons. The relative improvement observed with the proposed model increases with a longer prediction horizon, confirming the suitability of exploiting weekly patterns.

Also note that if an interval of 5 min is considered to evaluate prediction models against GEH, all current models fail to meet the standard for modelling by a significant margin, i.e. around 20% on average. However, averaging the output over 15 min smooths the data by removing the typical small oscillations in the flow over a short time duration. The resulting GEH value is then able to meet or exceed the modelling quality standards by an average of around 10%. It is suggested that GEH becomes a measure of choice not only for researchers involved in traffic modelling but also for researchers proposing models to predict traffic flows. Ideally, prediction models should meet the GEH standard even without additional filtering.

V. Conclusion

In this work, we have proposed a short-term traffic forecasting model based on a wide and deep architecture, which processes both short term spatial-temporal inputs and long-term periodic inputs. The short-term spatial-temporal input is processed by a Conv-LSTM block while the long-term periodic input is handled by a fully connected neural network. We find that by including only the relevant periodic data, i.e., weekly periodic data in our traffic forecasting model we surpass the prediction performance of existing stand-alone Conv-LSTM models whilst outperforming more complex composite deep learning models which incorporate all periodic features. Similar to [37], we have only considered networks in which the detectors are placed in succession. As a future work, a more complex and extensive road network with junctions and ramps should be considered and deep learning models which can handle the intricacies of such networks should be investigated. It is recommended to use, in addition to standard metrics, the traffic flow modelling criteria GEH to ensure that the model prediction quality is suitable for practitioners.

References

[1] T. J. Laetz, “Predictions and perceptions. Defining the traffic congestion problem,” *Technological Forecasting and Social Change*, vol. 38, no. 3, pp. 287–292, 1990, doi: 10.1016/0040-1625(90)90074-6.

[2] E. Manley and T. Cheng, “Understanding road congestion as an emergent property of traffic networks,” 2010.

[3] ATRI, “Trucking Industry Congestion Costs Now Top $74 Billion Annually - American Transportation Research Institute,” 2018. https://truckingresearch.org/2018/10/18/trucking-industry-congestion-costs-now-top-74-billion-annually/ (accessed May 25, 2020).

[4] J. S. Angarita-Zapata, A. D. Masegosa, and I. Triguero, “A Taxonomy of Traffic Forecasting Regression Problems from a Supervised Learning Perspective,” *IEEE Access*, vol. 7, pp. 68185–68205, 2019, doi: 10.1109/ACCESS.2019.2917228.

[5] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, “A comprehensive survey on traffic prediction,” *arXiv*. arXiv, Apr. 18, 2020.

[6] J. M. Hoque *et al.*, “The changing accuracy of traffic forecasts,” *Transportation (Amst)*, pp. 1–22, Feb. 2021, doi: 10.1007/S11116-021-10182-8/FIGURES/5.

[7] P. Xie, T. Li, J. Liu, S. Du, X. Yang, and J. Zhang, “Urban flow prediction from spatiotemporal data using machine learning: A survey,” *Information Fusion*, vol. 59, pp. 1–12, Jul. 2020, doi: 10.1016/j.inffus.2020.01.002.

[8] H. Yuan and G. Li, “A Survey of Traffic Prediction: from Spatio-Temporal Data to Intelligent Transportation,” *Data Science and Engineering*, vol. 6, no. 1, pp. 63–85, Mar. 2021, doi: 10.1007/s41019-020-00151-z.

[9] J. Guo and B. M. Williams, “Real-time short-term traffic speed level forecasting and uncertainty quantification using layered Kalman filters,” *Transportation Research Record*, no. 2175, pp. 28–37, 2010, doi: 10.3141/2175-04.

[10] N. Barimani, A. R. Kian, and B. Moshiri, “Real time adaptive non-linear estimator/predictor design for traffic systems with inadequate detectors,” *IET Intelligent Transport Systems*, vol. 8, no. 3, pp. 308–321, 2014, doi: 10.1049/IET-ITS.2013.0053/CITE/REFWORKS.

[11] S. R. Chandra and H. Al-Deek, “Predictions of freeway traffic speeds and volumes using vector autoregressive models,” *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 13, no. 2, pp. 53–72, Apr. 2009, doi: 10.1080/15472450902858368.

[12] B. M. Williams and L. A. Hoel, “Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results,” *Journal of Transportation Engineering*, 2003, doi: 10.1061/(ASCE)0733-947X(2003)129:6(664).

[13] T. Alghamdi, K. Elgazzar, M. Bayoumi, T. Sharaf, and S. Shah, “Forecasting traffic congestion using ARIMA modeling,” in *2019 15th International Wireless Communications and Mobile Computing Conference, IWCMC 2019*, 2019, pp. 1227–1232. doi: 10.1109/IWCMC.2019.8766698.

[14] E. Castillo, J. M. Menéndez, and S. Sánchez-Cambronero, “Predicting traffic flow using Bayesian networks,” *Transportation Research Part B: Methodological*, 2008, doi: 10.1016/j.trb.2007.10.003.

[15] X. Fei, C. C. Lu, and K. Liu, “A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction,” *Transportation Research Part C: Emerging Technologies*, 2011, doi: 10.1016/j.trc.2010.10.005.

[16] E. Bolshinsky and R. Freidman, “Traffic Flow Forecast Survey,” *Technion–Israel Institute of Technology.–2012.–Technical Report.–15*, pp. 1–15, 2012, [Online]. Available: http://nwwwn.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2012/CS/CS-2012-06.pdf

[17] A. Ermagun and D. Levinson, “Spatiotemporal traffic forecasting: review and proposed directions,” *Transport Reviews*, vol. 38, no. 6, pp. 786–814, Nov. 2018, doi: 10.1080/01441647.2018.1442887.

[18] Z. Zhu, B. Peng, C. Xiong, and L. Zhang, “Short-term traffic flow prediction with linear conditional Gaussian Bayesian network,” *Journal of Advanced Transportation*, vol. 50, no. 6, pp. 1111–1123, Oct. 2016, doi: 10.1002/ATR.1392.

[19] Y. Zhang and Y. Zhang, "A Comparative Study of Three Multivariate Short-Term Freeway Traffic Flow Forecasting Methods With Missing Data," *http://dx.doi.org/10.1080/15472450.2016.1147813*, vol. 20, no. 3, pp. 205–218, May 2016, doi: 10.1080/15472450.2016.1147813.
[20] B. Sun, W. Cheng, P. Goswami, and G. Bai, "Flow-aware WPT k-nearest neighbours regression for short-term traffic prediction," in *Proceedings - IEEE Symposium on Computers and Communications*, Sep. 2017, pp. 48–53. doi: 10.1109/ISCC.2017.8024503.
[21] Z. Liu, J. Guo, J. Cao, Y. U. N. Wei, and W. E. I. Huang, "A hybrid short-term traffic flow forecasting method based on neural networks combined with K-nearest neighbor," *Promet - Traffic - Traffico*, vol. 30, no. 4, pp. 445–456, Aug. 2018, doi: 10.7307/ptt.v30i4.2651.
[22] B. Sun, W. Cheng, P. Goswami, and G. Bai, "Short-term traffic forecasting using self-adjusting k-nearest neighbours," *IET Intelligent Transport Systems*, vol. 12, no. 1, pp. 41–48, Feb. 2018, doi: 10.1049/iet-its.2016.0263.
[23] W. C. Hong, Y. Dong, F. Zheng, and S. Y. Wei, "Hybrid evolutionary algorithms in a SVR traffic flow forecasting model," *Applied Mathematics and Computation*, vol. 217, no. 15, pp. 6733–6747, 2011, doi: 10.1016/j.amc.2011.01.073.
[24] W. C. Hong, "Application of seasonal SVR with chaotic immune algorithm in traffic flow forecasting," *Neural Computing and Applications*, vol. 21, no. 3, pp. 583–593, 2012, doi: 10.1007/s00521-010-0456-7.
[25] J. Gilmore and N. Abe, "Neural networks system for traffic congestion forecasting," in *Proceedings of the International Joint Conference on Neural Networks*, 1993, vol. 2, pp. 2025–2028. doi: 10.1109/ijcnn.1993.717056.
[26] F. Jin and S. Sun, "Neural network multitask learning for traffic flow forecasting," 2008. doi: 10.1109/IJCNN.2008.4634057.
[27] V. Osipov and D. Miloserdov, "Neural Network Forecasting of Traffic Congestion," in *Communications in Computer and Information Science*, 2019, vol. 1038 CCIS, pp. 248–254. doi: 10.1007/978-3-030-37858-5_20.
[28] H. Jiang, Y. Zou, S. Zhang, J. Tang, and Y. Wang, "Short-Term Speed Prediction Using Remote Microwave Sensor Data: Machine Learning versus Statistical Model," *Mathematical Problems in Engineering*, vol. 2016, 2016, doi: 10.1155/2016/9236156.
[29] Y. J. Wu, F. Chen, C. T. Lu, and S. Yang, "Urban Traffic Flow Prediction Using a Spatio-Temporal Random Effects Model," *http://dx.doi.org/10.1080/15472450.2015.1072050*, vol. 20, no. 3, pp. 282–293, May 2015, doi: 10.1080/15472450.2015.1072050.
[30] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic Flow Prediction with Big Data: A Deep Learning Approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, Apr. 2015, doi: 10.1109/TITS.2014.2345663.
[31] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014, doi: 10.1109/TITS.2014.2311123.
[32] Z. Zhao, W. Chen, X. Wu, P. C. V. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Image Processing*, 2017, doi: 10.1049/iet-its.2016.0208.
[33] Y. Zhong, X. Xie, J. Guo, Q. Wang, and S. Ge, "A new method for short-term traffic congestion forecasting based on LSTM," in *IOP Conference Series: Materials Science and Engineering*, 2018, vol. 383, no. 1. doi: 10.1088/1757-899X/383/1/012043.
[34] N. Ranjan, S. Bhandari, H. P. Zhao, H. Kim, and P. Khan, "City-wide traffic congestion prediction based on CNN, LSTM and transpose CNN," *IEEE Access*, vol. 8, pp. 81606–81620, 2020, doi: 10.1109/ACCESS.2020.2991462.
[35] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors (Switzerland)*, vol. 17, no. 4, Apr. 2017, doi: 10.3390/S17040818.
[36] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transportation Research Part C: Emerging Technologies*, vol. 90, 2018, doi: 10.1016/j.trc.2018.03.001.
[37] H. Zheng, F. Lin, X. Feng, and 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*, vol. 22, no. 11, pp. 6910–6920, Nov. 2021, doi: 10.1109/TITS.2020.2997352.
[38] Z. Cheng, J. Lu, H. Zhou, Y. Zhang, and L. Zhang, "Short-Term Traffic Flow Prediction: An Integrated Method of Econometrics and Hybrid Deep Learning," *IEEE Transactions on Intelligent Transportation Systems*, 2021, doi: 10.1109/TITS.2021.3052796.
[39] Y. Liu, H. Zheng, X. Feng, and Z. Chen, "Short-term traffic flow prediction with Conv-LSTM," *2017 9th International Conference on Wireless Communications and Signal Processing, WCSP 2017 - Proceedings*, vol. 2017-January, pp. 1–6, Dec. 2017, doi: 10.1109/WCSP.2017.8171119.
[40] H.-T. Cheng *et al.*, "Wide & Deep Learning for Recommender Systems," Jun. 2016, Accessed: Jan. 31, 2022. [Online]. Available: https://arxiv.org/abs/1606.07792v1
[41] Caltrans, "Performance measurement system (PeMS)," 2020. www.pems.dot.ca.gov/ (accessed Apr. 20, 2020).
[42] Jason Brownlee, *Introduction to Time Series Forecasting with Python*. 2020.
[43] Z. Zheng, Y. Yang, X. Niu, H. N. Dai, and Y. Zhou, "Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606–1615, Apr. 2018, doi: 10.1109/TII.2017.2785963.
[44] Z. Wang, W. Yan, and T. Oates, "Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2017-May, pp. 1578–1585, Nov. 2016, doi: 10.1109/IJCNN.2017.7966039.
[45] J. Bergstra and B. Yoshua, "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research*, vol. 13, no. 10, pp. 281–305, 2012, Accessed: Feb. 06, 2022. [Online]. Available: http://jmlr.org/papers/v13/bergstra12a.html
[46] Transport for London, "Traffic Modelling Guidelines Version 4," London, U.K., Sep. 2021.

MARTIN O. ESUGO is currently working towards his Ph.D. degree in computer science at the Institute for Clean Growth and Future Mobility, Coventry University. His current research interest includes intelligent transport systems, machine learning and deep learning.

Qian Lu received a PhD degree in control system engineering from the University of Manchester. She is currently an Assistant Professor of Connected and Autonomous Vehicles with Coventry University, UK. From 2016 to 2020, She was the lead system engineer of Assisted and Automated Driving System at Jaguar Land Rover, UK. Before joining JLR, she worked as a research fellow from 2013 to 2015 with the University of Surrey on the research of electric vehicle torque vectoring control. She also worked as a research associate with the University of Sheffield in the area of wind turbine blade control. Her current research includes automated parking systems, vehicle dynamic control, simulation and validation of autonomous vehicles.

Olivier C. L. Haas received his PhD from Coventry University (UK) in 1997. He is Associate Professor in Applied Control Systems and heads the ITS and 5G group within the Centre for Future Transport and Cities. His research interests include multi-objective optimization, control engineering, image processing and artificial intelligence. He is principal investigator for a commercial project on using deep learning for automated annotation for Advanced Driver Assistance Systems. He is reviewer for EPSRC (UK) and Plan Cancer (Fr). He has authored and co-authored over 150 scientific and technical papers, chapters and books and has successfully supervised 18 PhD students. https://pureportal.coventry.ac.uk/en/persons/olivier-haas